Amazon Elasticsearch Service

Developer Guide

API Version 2015-01-01

Amazon Elasticsearch Service: Developer Guide

Table of Contents

What Is Amazon Elasticsearch Service?

Amazon Elasticsearch Service (Amazon ES) is a managed service that makes it easy to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. Elasticsearch is a popular open-source search and analytics engine for use cases such as log analytics, real-time application monitoring, and clickstream analysis. With Amazon ES, you get direct access to the Elasticsearch APIs; existing code and applications work seamlessly with the service.

Amazon ES provisions all the resources for your Elasticsearch cluster and launches it. It also automatically detects and replaces failed Elasticsearch nodes, reducing the overhead associated with self-managed infrastructures. You can scale your cluster with a single API call or a few clicks in the console.

To get started using Amazon ES, you create a *domain*. An Amazon ES domain is synonymous with an Elasticsearch cluster. Domains are clusters with the settings, instance types, instance counts, and storage resources that you specify.

You can use the Amazon ES console to set up and configure a domain in minutes. If you prefer programmatic access, you can use the AWS CLI or the AWS SDKs.

Topics

Features of Amazon Elasticsearch Service

Amazon ES includes the following features:

Scale

- Numerous configurations of CPU, memory, and storage capacity, known as *instance types*
- Up to 1.5 PB of instance storage
- Amazon EBS storage volumes

Security

- AWS Identity and Access Management (IAM) access control
- Easy integration with Amazon VPC and VPC security groups
- Encryption of data at rest
- Amazon Cognito authentication for Kibana

Stability

- Multiple geographical locations for your resources, known as *regions* and *Availability Zones*
- Dedicated master nodes to offload cluster management tasks
- Automated snapshots to back up and restore Amazon ES domains
- Cluster node allocation across two Availability Zones in the same region, known as *zone awareness*

Integration with Popular Services

- Data visualization using Kibana
- Integration with Amazon CloudWatch for monitoring Amazon ES domain metrics and setting alarms
- Integration with AWS CloudTrail for auditing configuration API calls to Amazon ES domains
- Integration with Amazon S3, Amazon Kinesis, and Amazon DynamoDB for loading streaming data into Amazon ES

Supported Elasticsearch Versions

Amazon ES currently supports the following Elasticsearch versions:

- 6.3
- 6.2
- 6.0
- 5.6
- 5.5
- 5.3
- 5.1
- 2.3
- 1.5

Compared to earlier versions of Elasticsearch, the 6.*x* versions offer powerful features that make them faster, more secure, and easier to use. Here are some highlights:

- **Index splitting** – If an index outgrows its original number of shards, the `_split` API offers a convenient way to split each primary shard into two or more shards in a new index.
- **Vega visualizations** – Kibana 6.2 and newer support the Vega visualization language, which lets you make context-aware Elasticsearch queries, combine multiple data sources into a single graph, add user interactivity to graphs, and much more.
- **Ranking evaluation** – The `_rank_eval` API lets you measure and track how ranked search results perform against a set of queries to ensure that your searches perform as expected.
- **Composite aggregations** – These aggregations build composite buckets from one or more fields and sort them in "natural order" (alphabetically for terms, numerically or by date for histograms).
- **Higher indexing performance** – Newer versions of Elasticsearch provide superior indexing capabilities that significantly increase the throughput of data updates.
- **Better safeguards** – The 6.*x* versions of Elasticsearch offer many safeguards that are designed to prevent overly broad or complex queries from negatively affecting the performance and stability of the cluster.
- **Kibana autocomplete** – Kibana 6.3 and newer support autocomplete for queries, which greatly improves the day-to-day user experience.

For more information about the differences between Elasticsearch versions and the APIs that Amazon ES supports, see the section called “Supported Elasticsearch Operations” (p. 151).

If you start a new Elasticsearch project, we strongly recommend that you choose the latest supported Elasticsearch version. If you have an existing domain that uses an older Elasticsearch version, you can choose to keep the domain or migrate your data. For more information, see *Upgrading Elasticsearch* (p. 94).

Pricing for Amazon Elasticsearch Service

With AWS, you pay only for what you use. For Amazon ES, you pay for each hour of use of an EC2 instance and for the cumulative size of any EBS storage volumes attached to your instances. Standard AWS data transfer charges also apply.

However, a notable data transfer exception exists. If you use zone awareness (p. 43), Amazon ES does not bill for traffic between the two Availability Zones in which the domain resides. Significant data transfer occurs within a domain during shard allocation and rebalancing. Amazon ES neither meters nor bills for this traffic.

For full pricing details, see Amazon Elasticsearch Service Pricing. For information about charges incurred during configuration changes, see the section called “Charges for Configuration Changes” (p. 43).

If you qualify for the AWS Free Tier, you receive up to 750 hours per month of use with the `t2.micro.elasticsearch` or `t2.small.elasticsearch` instance types. You also receive up to 10 GB of Amazon EBS storage (Magnetic or General Purpose). For more information, see AWS Free Tier.

> **Note**
> Throughout this guide, 1 MB refers to 1024^2 or 1,048,576 bytes. Likewise, 1 GB refers to 1024^3 or 1,073,741,824 bytes.

Getting Started with Amazon Elasticsearch Service

To get started, sign up for an AWS account if you don't already have one. After you are set up with an account, complete the Getting Started (p. 5) tutorial for Amazon Elasticsearch Service. Consult the following introductory topics if you need more information while learning about the service:

- Create a domain (p. 10)
- Size your domain appropriately (p. 125)
- Control access to your domain (p. 30)
- Index data manually (p. 59) or from other services (p. 70)
- Use Kibana (p. 99) to search your data

Related Services

Amazon ES commonly is used with the following services:

AWS CloudTrail

Use AWS CloudTrail to get a history of the Amazon ES configuration API calls and related events for your account. For more information, see Logging Amazon Elasticsearch Service Configuration API Calls with AWS CloudTrail (p. 52).

Amazon CloudWatch

An Amazon ES domain automatically sends metrics to Amazon CloudWatch so that you can gather and analyze performance statistics. For more information, see Monitoring Cluster Metrics and Statistics with Amazon CloudWatch (Console) (p. 46).

CloudWatch Logs can also go the other direction. You might configure CloudWatch Logs to stream data to Amazon ES for analysis. To learn more, see the section called "Loading Streaming Data into Amazon ES from Amazon CloudWatch" (p. 81).

Amazon Kinesis

Kinesis is a managed service that scales elastically for real-time processing of streaming data at a massive scale. For more information, see the section called "Loading Streaming Data into Amazon ES from Amazon Kinesis Data Streams" (p. 75) and the section called "Loading Streaming Data into Amazon ES from Amazon Kinesis Data Firehose" (p. 80).

Amazon S3

Amazon Simple Storage Service (Amazon S3) provides storage for the internet. Amazon ES provides Lambda sample code for integration with Amazon S3. For more information, see the section called "Loading Streaming Data into Amazon ES from Amazon S3" (p. 70).

AWS IAM

AWS Identity and Access Management (IAM) is a web service that you can use to manage for your Amazon ES domains. For more information, see the IAM documentation and *Access Control* (p. 30).

AWS Lambda

AWS Lambda is a compute service that lets you run code without provisioning or managing servers. Amazon ES provides Lambda sample code to stream data from DynamoDB, Amazon S3, and Kinesis. For more information, see *Loading Streaming Data into Amazon ES* (p. 70).

Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. To learn more, see the section called "Loading Streaming Data into Amazon ES from Amazon DynamoDB" (p. 78).

Getting Started with Amazon Elasticsearch Service

This tutorial shows you how to use Amazon Elasticsearch Service (Amazon ES) to create and configure a test domain. An Amazon ES domain is synonymous with an Elasticsearch cluster. Domains are clusters with the settings, instance types, instance counts, and storage resources that you specify.

The tutorial walks you through the basic steps to get a domain up and running quickly. For more detailed information, see *Creating and Configuring Amazon ES Domains* (p. 10) and the other topics within this guide.

You can complete the following steps by using the Amazon ES console, the AWS CLI, or the AWS SDK:

1. Create an Amazon ES domain (p. 5)
2. Upload data to an Amazon ES domain for indexing (p. 7)
3. Search documents in an Amazon ES domain (p. 8)
4. Delete an Amazon ES domain (p. 9)

For information about installing and setting up the AWS CLI, see the AWS Command Line Interface User Guide.

Step 1: Create an Amazon ES Domain

Important
This process is a concise tutorial for configuring a *test domain*. It should not be used to create production domains. For a comprehensive version of the same process, see *Creating and Configuring Amazon ES Domains* (p. 10).

An Amazon ES domain is synonymous with an Elasticsearch cluster. Domains are clusters with the settings, instance types, instance counts, and storage resources that you specify. You can create an Amazon ES domain by using the console, the AWS CLI, or the AWS SDKs.

To create an Amazon ES domain (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. On the **Define domain** page, for **Elasticsearch domain name**, type a name for the domain. In this Getting Started tutorial, we use the domain name *movies* for the examples that we provide later in the tutorial.
4. For **Version**, choose an Elasticsearch version for your domain. We recommend that you choose the latest supported version. For more information, see the section called "Supported Elasticsearch Versions" (p. 2).
5. Choose **Next**.
6. For **Instance count**, choose the number of instances that you want. For this tutorial, you can use the default value of 1.
7. For **Instance type**, choose an instance type for the Amazon ES domain. For this tutoral, we recommend `t2.small.elasticsearch`, a small and inexpensive instance type suitable for testing purposes.

8. For now, you can ignore the **Enable dedicated master** and **Enable zone awareness** check boxes. For more information about both, see About Dedicated Master Nodes (p. 128) and Enabling Zone Awareness (p. 43).
9. For **Storage type**, choose **EBS**.

   a. For **EBS volume type**, choose General Purpose (SSD). For more information, see Amazon EBS Volume Types.

   b. For **EBS volume size**, type the size in GB of the external storage for *each* data node. For this tutorial, you can use the default value of 10.
10. For now, you can ignore **Enable encryption at rest**. For more information about the feature, see the section called "Encryption at Rest" (p. 134).
11. For **Automated snapshot start hour**, use the default value. For more information, see the section called "Configuring Automatic Snapshots" (p. 23).
12. Choose **Next**.
13. For simplicity in this tutorial, we recommend an IP-based access policy. On the **Set up access** page, in the **Network configuration** section, choose **Public access**.
14. For now, you can ignore **Kibana authentication**. For more information about the feature, see *Authentication for Kibana* (p. 104).
15. For **Set the domain access policy to**, choose **Allow access to the domain from specific IP(s)** and enter your public IP address, which you can find by searching for "What is my IP?" on most search engines. Then choose **OK**.

    To learn more about public access, VPC access, and access policies in general, see *Access Control* (p. 30) and *VPC Support* (p. 117).
16. Choose **Next**.
17. On the **Review** page, review your domain configuration, and then choose **Confirm**.

    Note
    New domains take roughly ten minutes to initialize. After your domain is initialized, you can upload data and make changes to the domain.

To create an Amazon ES domain (AWS CLI)

- Run the following command to create an Amazon ES domain.

  The command creates a domain named *movies* with Elasticsearch version 6.0. It specifies one instance of the `t2.small.elasticsearch` instance type. The instance type requires EBS storage, so it specifies a 10 GB volume. Finally, the command applies an IP-based access policy that restricts access to the domain to a single IP address.

  You need to replace `your_ip_address` in the command with your public IP address, which you can find by searching for "What is my IP?" on Google.

```
aws es create-elasticsearch-domain --domain-name movies --elasticsearch-version 6.0
 --elasticsearch-cluster-config InstanceType=t2.small.elasticsearch,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 --access-
policies '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"*"},"Action":["es:*"],"Condition":{"IpAddress":{"aws:SourceIp":
["your_ip_address"]}}}]}'
```

  Note
  New domains take roughly ten minutes to initialize. After your domain is initialized, you can upload data and make changes to the domain.

Use the following command to query the status of the new domain:

```
aws es describe-elasticsearch-domain --domain movies
```

To create an Amazon ES domain (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon ES Configuration API Reference (p. 167), including the `CreateElasticsearchDomain` action. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Step 2: Upload Data to an Amazon ES Domain for Indexing

> **Important**
> This process is a concise tutorial for uploading a small amount of test data. For more information, see *Indexing Data* (p. 59).

You can upload data to an Amazon Elasticsearch Service domain for indexing using the Elasticsearch index and bulk APIs from the command line.

- Use the index API to add or update a single Elasticsearch document.
- Use the bulk API to add or update multiple Elasticsearch documents that are described in the same JSON file.

The following example requests use curl, a common HTTP client, for brevity and convenience. Clients like curl can't perform the request signing that is required if your access policies specify IAM users or roles. To successfully perform the instructions in this step, you must use an IP address-based access policy that allows unauthenticated access, like you configured in step 1 (p. 5).

You can install curl on Windows and use it from the command prompt, but Windows users might find it more convenient to use a tool like Cygwin or the Windows Subsystem for Linux. macOS and most Linux distributions come with curl pre-installed.

To upload a single document to an Amazon ES domain

- Run the following command to add a single document to the *movies* domain:

```
curl -XPUT elasticsearch_domain_endpoint/movies/_doc/1 -d '{"director": "Burton,
 Tim", "genre": ["Comedy","Sci-Fi"], "year": 1996, "actor": ["Jack Nicholson","Pierce
 Brosnan","Sarah Jessica Parker"], "title": "Mars Attacks!"}' -H 'Content-Type:
 application/json'
```

For a detailed explanation of this command and how to make signed requests to Amazon ES, see *Indexing Data* (p. 59).

To upload a JSON file that contains multiple documents to an Amazon ES domain

1. Create a file called `bulk_movies.json`. Copy and paste the following content into it, and add a trailing newline:

```
{ "index" : { "_index": "movies", "_type" : "_doc", "_id" : "2" } }
{"director": "Frankenheimer, John", "genre": ["Drama", "Mystery", "Thriller"], "year":
 1962, "actor": ["Lansbury, Angela", "Sinatra, Frank", "Leigh, Janet", "Harvey,
 Laurence", "Silva, Henry", "Frees, Paul", "Gregory, James", "Bissell, Whit", "McGiver,
 John", "Parrish, Leslie", "Edwards, James", "Flowers, Bess", "Dhiegh, Khigh", "Payne,
 Julie", "Kleeb, Helen", "Gray, Joe", "Nalder, Reggie", "Stevens, Bert", "Masters,
 Michael", "Lowell, Tom"], "title": "The Manchurian Candidate"}
{ "index" : { "_index": "movies", "_type" : "_doc", "_id" : "3" } }
{"director": "Baird, Stuart", "genre": ["Action", "Crime", "Thriller"], "year": 1998,
 "actor": ["Downey Jr., Robert", "Jones, Tommy Lee", "Snipes, Wesley", "Pantoliano,
 Joe", "Jacob, Ir\u00e8ne", "Nelligan, Kate", "Roebuck, Daniel", "Malahide, Patrick",
 "Richardson, LaTanya", "Wood, Tom", "Kosik, Thomas", "Stellate, Nick", "Minkoff,
 Robert", "Brown, Spitfire", "Foster, Reese", "Spielbauer, Bruce", "Mukherji, Kevin",
 "Cray, Ed", "Fordham, David", "Jett, Charlie"], "title": "U.S. Marshals"}
{ "index" : { "_index": "movies", "_type" : "_doc", "_id" : "4" } }
{"director": "Ray, Nicholas", "genre": ["Drama", "Romance"], "year": 1955, "actor":
 ["Hopper, Dennis", "Wood, Natalie", "Dean, James", "Mineo, Sal", "Backus, Jim",
 "Platt, Edward", "Ray, Nicholas", "Hopper, William", "Allen, Corey", "Birch, Paul",
 "Hudson, Rochelle", "Doran, Ann", "Hicks, Chuck", "Leigh, Nelson", "Williams, Robert",
 "Wessel, Dick", "Bryar, Paul", "Sessions, Almira", "McMahon, David", "Peters Jr.,
 House"], "title": "Rebel Without a Cause"}
```

2. Run the following command to upload the file to the *movies* domain:

```
curl -XPOST elasticsearch_domain_endpoint/_bulk --data-binary @bulk_movies.json -H
 'Content-Type: application/json'
```

For more information about the bulk file format, see *Indexing Data* (p. 59).

> **Note**
> Amazon ES supports migrating data from manual snapshots taken on both Amazon ES and self-managed Elasticsearch clusters. Restoring a snapshot from a self-managed Elasticsearch cluster is a common way to migrate data to Amazon ES. For more information, see the section called “Restoring Snapshots” (p. 92).

Step 3: Search Documents in an Amazon ES Domain

To search documents in an Amazon Elasticsearch Service domain, use the Elasticsearch search API. Alternatively, you can use Kibana (p. 99) to search documents in the domain.

To search documents from the command line

- Run the following command to search the *movies* domain for the word *mars*:

```
curl -XGET 'elasticsearch_domain_endpoint/movies/_search?q=mars'
```

To search documents from an Amazon ES domain by using Kibana

1. Point your browser to the Kibana plugin for your Amazon ES domain. You can find the Kibana endpoint on your domain dashboard on the Amazon ES console. The URL follows the format of:

```
https://domain.region.es.amazonaws.com/_plugin/kibana/
```

2. To use Kibana, you must configure at least one index pattern. Kibana uses these patterns to identity which indices you want to analyze. For this tutorial, enter *movies* and choose **Create**.
3. The **Index Patterns** screen shows your various document fields, fields like `actor` and `director`. For now, choose **Discover** to search your data.

4. In the search bar, type *mars*, and then press **Enter**. Note how the similarity score (`_score`) increases when you search for the phrase *mars attacks*.

Step 4: Delete an Amazon ES Domain

Because the *movies* domain from this tutorial is for test purposes, you should delete it when you are finished experimenting to avoid incurring charges.

To delete an Amazon ES domain (console)

1. Log in to the **Amazon Elasticsearch Service** console.
2. In the navigation pane, under **My domains**, choose the *movies* domain.
3. Choose **Delete Elasticsearch domain**.
4. Choose **Delete domain.**
5. Select the **Delete the domain** check box, and then choose **Delete**.

To delete an Amazon ES domain (AWS CLI)

- Run the following command to delete the *movies* domain:

```
aws es delete-elasticsearch-domain --domain-name movies
```

Note
Deleting a domain deletes all billable Amazon ES resources. However, any manual snapshots of the domain that you created are not deleted. Consider saving a snapshot if you might need to recreate the Amazon ES domain in the future. If you don't plan to recreate the domain, you can safely delete any snapshots that you created manually.

To delete an Amazon ES domain (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon ES Configuration API Reference (p. 167), including the `DeleteElasticsearchDomain` action. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Creating and Configuring Amazon Elasticsearch Service Domains

This chapter describes how to create and configure Amazon Elasticsearch Service (Amazon ES) domains. An Amazon ES domain is synonymous with an Elasticsearch cluster. Domains are clusters with the settings, instance types, instance counts, and storage resources that you specify.

Unlike the brief instructions in the Getting Started (p. 5) tutorial, this chapter describes all options and provides relevant reference information. You can complete each procedure by using instructions for the Amazon ES console, the AWS Command Line Interface (AWS CLI), or AWS SDKs.

Topics

Creating Amazon ES Domains

This section describes how to create Amazon ES domains by using the Amazon ES console or by using the AWS CLI with the `create-elasticsearch-domain` command. The procedures for the AWS CLI include syntax and examples.

Creating Amazon ES Domains (Console)

Use the following procedure to create an Amazon ES domain by using the console.

To create an Amazon ES domain (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. Choose **Create a new domain**.

   Alternatively, choose **Get Started** if this is the first Amazon ES domain that you will create for your AWS account.
4. On the **Define domain** page, for **Domain name**, type a name for your domain. The domain name must meet the following criteria:

   - Uniquely identifies a domain
   - Starts with a lowercase letter
   - Contains between 3 and 28 characters

- Contains only lowercase letters a-z, the numbers 0-9, and the hyphen (-)

5. For **Version**, choose an Elasticsearch version for your domain. We recommend that you choose the latest version. For more information, see the section called “Supported Elasticsearch Versions” (p. 2).
6. Choose **Next**.
7. For **Instance count**, choose the number of instances that you want.

   The default is one. For maximum values, see the section called “Cluster and Instance Limits” (p. 225). We recommend a minimum of three instances to avoid potential Elasticsearch issues, such as the split brain issue. If you have three dedicated master nodes (p. 128), we still recommend a minimum of two data nodes for replication. Single node clusters are fine for development and testing, but should not be used for production workloads. For more guidance, see the section called “Sizing Amazon ES Domains” (p. 125).
8. For **Instance type**, choose an instance type for the data nodes.

   To see a list of the instance types that Amazon ES supports, see Supported Instance Types (p. 150).
9. (Optional) If you must ensure cluster stability or if you have a domain that has more than 10 instances, enable a dedicated master node. Dedicated master nodes increase cluster stability and are required for a domain that has an instance count greater than 10. For more information, see About Dedicated Master Nodes (p. 128).

   a. Select the **Enable dedicated master** check box.
   b. For **Dedicated master instance type**, choose an instance type for the dedicated master node.

      For a list of the instance types that Amazon ES supports, see Supported Instance Types (p. 150).

      Note

      - You can choose an instance type for the dedicated master node that differs from the instance type that you choose for the data nodes. For example, you might select general purpose or storage-optimized instances for your data nodes, but compute-optimized instances for your dedicated master nodes.

   c. For **Dedicated master instance count**, choose the number of instances for the dedicated master node.

      We recommend choosing an odd number of instances to avoid potential Elasticsearch issues, such as the split brain issue. The default and recommended number is three.
10. (Optional) Select the **Enable zone awareness** check box.

    Zone awareness distributes Amazon ES data nodes across two Availability Zones in the same region. If you enable zone awareness, you must have an even number of instances in the instance count, and you must use the native Elasticsearch API to create replica shards for your cluster. This process allows for the even distribution of shards across two Availability Zones. For more information, see Enabling Zone Awareness (p. 43).
11. For **Storage type**, choose either **Instance** (the default) or **EBS**.

    If your Amazon ES domain requires more storage, use an EBS volume for storage rather than the storage that is attached to the selected instance type. Domains with large indices or large numbers of indices often benefit from the increased storage capacity of EBS volumes. For guidance on creating especially large domains, see Petabyte Scale (p. 132). If you choose **EBS**, the following boxes appear:

    a. For **EBS volume type**, choose an EBS volume type.

       If you choose Provisioned IOPS (SSD) for the EBS volume type, for **Provisioned IOPS**, type the baseline IOPS performance that you want. For more information, see Amazon EBS Volumes in the Amazon EC2 documentation.

   b. For **EBS volume size**, type the size of the EBS volume that you want to attach to each data node.

      EBS volume size is per node. You can calculate the total cluster size for the Amazon ES domain using the following formula: (number of data nodes) * (EBS volume size). The minimum and maximum size of an EBS volume depends on both the specified EBS volume type and the instance type that it's attached to. To learn more, see EBS Volume Size Limits (p. 225).

12. (Optional) To enable encryption of data at rest, select the **Enable encryption at rest** check box.

    Select **(Default) aws/es** to have Amazon ES create a KMS encryption key on your behalf (or use the one that it already created). Otherwise, choose your own KMS encryption key from the **KMS master key** menu. To learn more, see the section called "Encryption at Rest" (p. 134).

13. For **Automated snapshot start hour**, choose the hour for automated daily snapshots of domain indices.

    For more information and recommendations, see the section called "Configuring Automatic Snapshots" (p. 23).

14. (Optional) Choose **Advanced options**. For a summary of options, see the section called "Configuring Advanced Options" (p. 24).

15. Choose **Next**.

16. On the **Set up access** page, in the **Network configuration** section, choose either **Public Access** or **VPC access**. If you choose **Public access**, skip to step 17. If you choose **VPC access**, ensure that you have met the prerequisites (p. 121), and then do the following:

    a. For **VPC**, choose the ID of the VPC that you want to use.

       Note
       The VPC and domain must be in the same AWS Region, and you must select a VPC with tenancy set to **Default**. Amazon ES does not yet support VPCs that use dedicated tenancy.

    b. For **Subnet**, choose a subnet. If you enabled zone awareness in step 10, you must choose two subnets. Amazon ES will place a VPC endpoint and *elastic network interfaces* (ENIs) in the subnet or subnets.

       Note
       You must reserve sufficient IP addresses for the network interfaces in the subnet (or subnets). For more information, see Reserving IP Addresses in a VPC Subnet (p. 122).

    c. For **Security groups**, choose the VPC security groups that need access to the Amazon ES domain. For more information, see *VPC Support* (p. 117).

    d. For **IAM role**, keep the default role. Amazon ES uses this predefined role (also known as a *service-linked role*) to access your VPC and to place a VPC endpoint and network interfaces in the subnet of the VPC. For more information, see Service-Linked Role for VPC Access (p. 123).

17. (Optional) If you want to enable node-to-node encryption (p. 136), choose **Node-to-node encryption**.

18. (Optional) If you want to protect Kibana with a login page, choose **Enable Amazon Cognito for authentication**.

    - Choose the Amazon Cognito user pool and identity pool that you want to use for Kibana authentication. For guidance on creating these resources, see *Authentication for Kibana* (p. 104).

19. For **Set the domain access policy to**, choose a preconfigured policy from the **Select a template** dropdown list and edit it to meet the needs of your domain. Alternatively, you can add one or more Identity and Access Management (IAM) policy statements in the **Add or edit the access policy** box. For more information, see *Access Control* (p. 30), the section called "Configuring Access Policies" (p. 21), and the section called "About Access Policies on VPC Domains" (p. 119).

> **Note**
> If you chose **VPC access** in step 16, the IP-based policy template is not available in the dropdown list, and you can't configure an IP-based policy manually. Instead, you can use security groups to control which IP addresses can access the domain. To learn more, see the section called “About Access Policies on VPC Domains” (p. 119).

20. Choose **Next**.
21. On the **Review** page, review your domain configuration, and then choose **Confirm and create**.
22. Choose **OK**.

> **Note**
> New domains take up to ten minutes to initialize. After your domain is initialized, you can upload data and make changes to the domain.

Creating Amazon ES Domains (AWS CLI)

Instead of creating an Amazon ES domain by using the console, you can create a domain by using the AWS CLI. Use the following syntax to create an Amazon ES domain.

```
aws es create-elasticsearch-domain --domain-name <value>

  [--elasticsearch-version <value>]
  [--elasticsearch-cluster-config <value>]
  [--ebs-options <value>]
  [--access-policies <value>]
  [--snapshot-options <value>]
  [--vpc-options <value>]
  [--advanced-options <value>]
  [--log-publishing-options <value>]
  [--cli-input-json <value>]
  [--generate-cli-skeleton <value>]
  [--encryption-at-rest-options <value>]
  [--cognito-options <value>]
  [--node-to-node-encryption-options <value>]
```

The following table provides more information about each of the optional parameters.

Optional Parameter	Description
`--elasticsearch-version`	Specifies the Elasticsearch version of the domain. If not specified, the default value is 1.5. For more information, see Choosing an Elasticsearch Version (p. 2).
`--elasticsearch-cluster-config`	Specifies the instance type and count of the domain, whether zone awareness is enabled, and whether the domain uses a dedicated master node. Dedicated master nodes increase cluster stability and are required for a domain that has an instance count greater than 10. For more information, see Configuring Amazon ES Domains (p. 17).
`--ebs-options`	Specifies whether the domain uses an EBS volume for storage. If true, this parameter must also specify the EBS volume type, size, and, if applicable, IOPS value. For more information, see Configuring EBS-based Storage (p. 19).
`--access-policies`	Specifies the access policy for the domain. For more information, see Configuring Access Policies (p. 22).

Optional Parameter	**Description**
`--snapshot-options`	Specifies the hour in UTC during which the service performs a daily automated snapshot of the indices in the domain. The default value is `0`, or midnight, which means that the snapshot is taken anytime between midnight and 1:00 AM. For more information, see Configuring Snapshots (p. 23).
`--advanced-options`	Specifies whether to allow references to indices in the bodies of HTTP request objects. For more information, see Configuring Advanced Options (p. 25).
`--generate-cli-skeleton`	Displays JSON for all specified parameters. Save the output to a file so that you can later read the file with the `--cli-input-json` parameter rather than typing the parameters at the command line. For more information, see Generate CLI Skeleton and CLI Input JSON Parameters in the *AWS Command Line Interface User Guide*.
`--cli-input-json`	Specifies the name of a JSON file that contains a set of CLI parameters. For more information, see Generate CLI Skeleton and CLI Input JSON Parameters in the *AWS Command Line Interface User Guide*.
`--log-publishing-options`	Specifies whether Amazon ES should publish Elasticsearch logs to CloudWatch. For more information, see the section called "Configuring Logs" (p. 25).
`--vpc-options`	Specifies whether to launch the Amazon ES domain within an Amazon VPC (VPC). To learn more, see *VPC Support* (p. 117).
`--encryption-at-rest-options`	Specifies whether to enable encryption of data at rest (p. 134).
`--cognito-options`	Specifies whether to use *Authentication for Kibana* (p. 104).
`--node-to-node-encryption-options`	Specify `true` to enable node-to-node encryption (p. 136).

Examples

The first example demonstrates the following Amazon ES domain configuration:

- Creates an Amazon ES domain named *weblogs* with Elasticsearch version 5.5
- Populates the domain with two instances of the m4.large.elasticsearch instance type
- Uses a 100 GB Magnetic disk EBS volume for storage for each data node
- Allows anonymous access, but only from a single IP address: 192.0.2.0/32

```
aws es create-elasticsearch-domain --domain-name weblogs --elasticsearch-version 5.5 --
elasticsearch-cluster-config  InstanceType=m4.large.elasticsearch,InstanceCount=2 --ebs-
options EBSEnabled=true,VolumeType=standard,VolumeSize=100 --access-policies '{"Version":
 "2012-10-17", "Statement": [{"Action": "es:*", "Principal":"*","Effect": "Allow",
 "Condition": {"IpAddress":{"aws:SourceIp":["192.0.2.0/32"]}}}]}'
```

The next example demonstrates the following Amazon ES domain configuration:

- Creates an Amazon ES domain named *weblogs* with Elasticsearch version 5.5

- Populates the domain with six instances of the m4.large.elasticsearch instance type
- Uses a 100 GB General Purpose (SSD) EBS volume for storage for each data node
- Restricts access to the service to a single user, identified by the user's AWS account ID: 555555555555
- Enables zone awareness

```
aws es create-elasticsearch-domain --domain-name weblogs --
elasticsearch-version 5.5 --elasticsearch-cluster-config
 InstanceType=m4.large.elasticsearch,InstanceCount=6,ZoneAwarenessEnabled=true
 --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 --access-policies
 '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": {"AWS":
 "arn:aws:iam::555555555555:root" }, "Action":"es:*", "Resource": "arn:aws:es:us-
east-1:555555555555:domain/logs/*" } ] }'
```

The next example demonstrates the following Amazon ES domain configuration:

- Creates an Amazon ES domain named *weblogs* with Elasticsearch version 5.5
- Populates the domain with ten instances of the m4.xlarge.elasticsearch instance type
- Populates the domain with three instances of the m4.large.elasticsearch instance type to serve as dedicated master nodes
- Uses a 100 GB Provisioned IOPS EBS volume for storage, configured with a baseline performance of 1000 IOPS for each data node
- Restricts access to a single user and to a single subresource, the `_search` API
- Configures automated daily snapshots of the indices for 03:00 UTC

```
aws es create-elasticsearch-domain --domain-name weblogs --
elasticsearch-version 5.5 --elasticsearch-cluster-config
 InstanceType=m4.xlarge.elasticsearch,InstanceCount=10,DedicatedMasterEnabled=true,DedicatedMasterType=
 --ebs-options EBSEnabled=true,VolumeType=io1,VolumeSize=100,Iops=1000 --access-
policies '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal":
 { "AWS": "arn:aws:iam::555555555555:root" }, "Action": "es:*", "Resource":
 "arn:aws:es:us-east-1:555555555555:domain/mylogs/_search" } ] }' --snapshot-options
 AutomatedSnapshotStartHour=3
```

> **Note**
> If you attempt to create an Amazon ES domain and a domain with the same name already exists, the CLI does not report an error. Instead, it returns details for the existing domain.

Creating Amazon ES Domains (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon ES Configuration API Reference (p. 167), including `CreateElasticsearchDomain`. For sample code, see *Using the AWS SDKs* (p. 163). For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring Amazon ES Domains

To meet the demands of increased traffic and data, you can update your Amazon ES domain configuration with any of the following changes:

- Change the instance count
- Change the instance type
- Enable or disable dedicated master nodes

- Enable or disable zone awareness
- Configure storage configuration
- Change the start time for automated snapshots of domain indices
- Change the VPC subnets and security groups
- Configure advanced options

> **Note**
> For information about configuring a domain to use an EBS volume for storage, see Configuring EBS-based Storage (p. 18).

Configuring Amazon ES Domains (Console)

Use the following procedure to update your Amazon ES configuration by using the console.

To configure an Amazon ES domain (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to update.
4. Choose **Configure cluster**.
5. On the **Configure cluster** page, update the configuration of the domain.

   The cluster is a collection of one or more data nodes, optional dedicated master nodes, and storage required to run Amazon ES and operate your domain.

   a. If you want to change the instance type for data nodes, for **Instance type**, choose a new instance type.

      To see a list of the instance types that Amazon ES supports, see Supported Instance Types (p. 150).
   b. If you want to change the instance count, for **Instance count**, choose an integer from one to twenty. To request an increase up to 100 instances per domain, create a case with the AWS Support Center.
   c. If you want to improve cluster stability or if your domain has an instance count greater than 10, enable a dedicated master node for your cluster. For more information, see About Dedicated Master Nodes (p. 128).

      i. Select the **Enable dedicated master** check box.
      ii. For **Dedicated master instance type**, choose an instance type for the dedicated master node.

          You can choose an instance type for the dedicated master node that differs from the instance type that you choose for the data nodes.

          To see a list of the instance types that Amazon ES supports, see Supported Instance Types (p. 150).
      iii. For **Dedicated master instance count**, choose the number of instances for the dedicated master node.

          We recommend choosing an odd number of instances to avoid potential Amazon ES issues, such as the split brain issue. The default and recommended number is three.
   d. If you want to enable zone awareness, select the **Enable zone awareness** check box. If you enable zone awareness, you must have an even number of instances in your instance count. This allows for the even distribution of shards across two Availability Zones in the same region.

e. If you want to change the hour during which the service takes automated daily snapshots of the primary index shards of your Amazon ES domain, for **Automated snapshot start hour**, choose an integer.

f. If you didn't enable VPC access when you created the domain, skip to step 7. If you enabled VPC access, you can change the subnet that the VPC endpoint is placed in, and you can change the security groups:

    i. For **Subnets**, choose a subnet. The subnet must have a sufficient number of IP addresses reserved for the network interfaces. If you enabled zone awareness, you must choose two subnets. The subnets must be in different Availability Zones in the same region. For more information, see *VPC Support* (p. 117).

    ii. For **Security groups**, add the security groups that need access to the domain.

g. (Optional) Choose **Advanced options**. For a summary of options, see the section called "Configuring Advanced Options" (p. 24)

h. Choose **Submit**.

Configuring Amazon ES Domains (AWS CLI)

Use the `elasticsearch-cluster-config` option to configure your Amazon ES cluster by using the AWS CLI. The following syntax is used by both the `create-elasticsearch-domain` and `update-elasticsearch-domain-config` commands.

Syntax

```
--elasticsearch-cluster-config
 InstanceType=<value>,InstanceCount=<value>,DedicatedMasterEnabled=<value>,DedicatedMasterType=<value>,
```

Note
Do not include spaces between parameters for the same option.

The following table describes the parameters in more detail.

Parameter	Valid Values	Description
`InstanceType`	Any supported instance type. See Supported Instance Types.	The hardware configuration of the computer that hosts the instance. The default is m4.large.elasticsearch.
`InstanceCount`	Integer	The number of instances in the Amazon ES domain. The default is one, and the maximum default limit is twenty. To request an increase up to 100 instances per domain, create a case with the AWS Support Center.
`DedicatedMasterEnabled`	`true` or `false`	Specifies whether to use a dedicated master node for the Amazon ES domain. The default value is `false`.
`DedicatedMasterType`	Any supported instance type	The hardware configuration of the computer that hosts the master node. The default is m4.large.elasticsearch.
`DedicatedMasterCount`	Integer	The number of instances used for the dedicated master node. The default is three.

Parameter	Valid Values	Description
`ZoneAwarenessEnabled`	`true` or `false`	Specifies whether to enable zone awareness for the Amazon ES domain. The default value is `false`.

Examples

The following example creates an Amazon ES domain named `mylogs` with Elasticsearch version 5.5 with two instances of the m4.large.elasticsearch instance type and zone awareness enabled:

```
aws es create-elasticsearch-domain --domain-name mylogs --
elasticsearch-version 5.5 --elasticsearch-cluster-config
 InstanceType=m4.large.elasticsearch,InstanceCount=2,DedicatedMasterEnabled=false,ZoneAwarenessEnabled=
```

However, you likely will want to reconfigure your new Amazon ES domain as network traffic grows and as the quantity and size of documents increase. For example, you might decide to use a larger instance type, use more instances, and enable a dedicated master node. The following example updates the domain configuration with these changes:

```
aws es update-elasticsearch-domain-config --domain-name mylogs --elasticsearch-cluster-
config
 InstanceType=m4.xlarge.elasticsearch,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedMasterType=m
```

Configuring Amazon ES Domains (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon ES Configuration API Reference (p. 167), including `UpdateElasticsearchDomainConfig`. For sample code, see *Using the AWS SDKs* (p. 163). For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring EBS-based Storage

An Amazon EBS volume is a block-level storage device that you can attach to a single instance. EBS volumes enable you to independently scale the storage resources of your Amazon ES domain from its compute resources. EBS volumes are most useful for domains with large datasets, but without the need for large compute resources. EBS volumes are much larger than the default storage provided by the instance. Amazon Elasticsearch Service supports the following EBS volume types:

- General Purpose (SSD)
- Provisioned IOPS (SSD)
- Magnetic

Note

When changing an EBS volume type from provisioned IOPS to non-provisioned EBS volume types, set the IOPS value to `0`.

Warning

Currently, if the data node that is attached to an EBS volume fails, the EBS volume also fails.

Configuring EBS-based Storage (Console)

Use the following procedure to enable EBS-based storage by using the console.

To enable EBS-based storage (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to configure.
4. Choose **Configure cluster**.
5. For **Storage type**, choose **EBS**.
6. For **EBS volume type**, choose an EBS volume type.

   If you choose **Provisioned IOPS (SSD)** for the EBS volume type, for **Provisioned IOPS**, type the baseline IOPS performance that you want.
7. For **EBS volume size**, type the size that you want for the EBS volume.

   EBS volume size is per node. You can calculate the total cluster size for the Amazon ES domain using the following formula: (number of data nodes) * (EBS volume size). The minimum and maximum size of an EBS volume depends on both the specified EBS volume type and the instance type to which it is attached. To learn more, see EBS Volume Size Limits (p. 225).
8. Choose **Submit**.

> **Note**
> Set the IOPS value for a Provisioned IOPS EBS volume to no more than 30 times the maximum storage of the volume. For example, if your volume has a maximum size of 100 GB, you can't assign an IOPS value for it that is greater than 3000.

For more information, see Amazon EBS Volumes in the Amazon EC2 documentation.

Configuring EBS-based Storage (AWS CLI)

Use the `--ebs-options` option to configure EBS-based storage by using the AWS CLI. The following syntax is used by both the `create-elasticsearch-domain` and `update-elasticsearch-domain-config` commands.

Syntax

```
--ebs-options EBSEnabled=<value>,VolumeType=<value>,VolumeSize=<value>,IOPS=<value>
```

Parameter	Valid Values	Description
`EBSEnabled`	`true` or `false`	Specifies whether to use an EBS volume for storage rather than the storage provided by the instance. The default value is `false`.
`VolumeType`	Any of the following: • `gp2` (General Purpose SSD) • `io1` (Provisioned IOPS SSD) • `standard` (Magnetic)	The EBS volume type to use with the Amazon ES domain.
`VolumeSize`	Integer	Specifies the size of the EBS volume for each data node in GB. The minimum and maximum size of an EBS volume depends on both the specified EBS volume type and the instance type to which it is attached. To see a

Parameter	Valid Values	Description
		table that shows the minimum and maximum EBS size for each instance type, see Service Limits.
`IOPS`	Integer	Specifies the baseline I/O performance for the EBS volume. This parameter is used only by Provisioned IOPS (SSD) volumes. The minimum value is 1000. The maximum value is 16000.

Note

We recommend that you do not set the IOPS value for a Provisioned IOPS EBS volume to more than 30 times the maximum storage of the volume. For example, if your volume has a maximum size of 100 GB, you should not assign an IOPS value for it that is greater than 3000. For more information, including use cases for each volume type, see Amazon EBS Volume Types in the Amazon EC2 documentation.

Examples

The following example creates a domain named `mylogs` with Elasticsearch version 5.5 with a 10 GB General Purpose EBS volume:

```
aws es create-elasticsearch-domain --domain-name=mylogs --elasticsearch-version 5.5 --ebs-
options EBSEnabled=true,VolumeType=gp2,VolumeSize=10
```

However, you might need a larger EBS volume as the size of your search indices increases. For example, you might opt for a 100 GB Provisioned IOPS volume with a baseline I/O performance of 3000 IOPS. The following example updates the domain configuration with those changes:

```
aws es update-elasticsearch-domain-config --domain-name=mylogs --ebs-options
 EBSEnabled=true,VolumeType=io1,VolumeSize=100,IOPS=3000
```

Configuring EBS-based Storage (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon ES Configuration API Reference (p. 167), including the `--ebs-options` parameter for `UpdateElasticsearchDomainConfig`. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Modifying VPC Access Configuration

If you configured a domain to reside within a VPC, you can modify the configuration using the Amazon ES console. To migrate a public domain to a VPC domain, see the section called “Migrating from Public Access to VPC Access” (p. 123).

Configuring VPC Access (Console)

Use the following procedure to configure VPC access by using the console.

To configure VPC access (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.

3. In the navigation pane, under **My domains**, choose the domain that you want to configure.
4. Choose **Configure cluster**.
5. In the **Network configuration** section, for **Subnets**, choose a subnet. If you enabled zone awareness, you must choose two subnets. The subnets must be in different Availability Zones in the same region. For more information, see *VPC Support* (p. 117).

   Note
   You must reserve sufficient IP addresses for the network interfaces in the subnet (or subnets). For more information, see the section called "Reserving IP Addresses in a VPC Subnet" (p. 122).
6. For **Security groups**, add the security groups that need access to the domain.
7. Choose **Submit**.

Configuring Amazon Cognito Authentication for Kibana

See *Authentication for Kibana* (p. 104).

Configuring Access Policies

Amazon Elasticsearch Service offers several ways to configure access to your Amazon ES domains. For more information, see *Access Control* (p. 30).

The console provides preconfigured access policies that you can customize for the specific needs of your domain. You also can import access policies from other Amazon ES domains. For information on how these access policies interact with VPC access, see the section called "About Access Policies on VPC Domains" (p. 119).

Configuring Access Policies (Console)

Use the following procedure to configure access policies by using the console.

To configure access policies (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to update.
4. Choose **Modify access policy**.
5. Edit the access policy.

   Alternatively, choose one of the policy templates from the **Select a template** dropdown list, and then edit it as needed for your domain.

Preconfigured Access Policy	Description
Allow or deny access to one or more AWS accounts or IAM users	Allows or denies access to one or more AWS accounts or IAM users or roles.
Allow access to the domain from specific IP(s)	This policy is used to restrict anonymous access to a specific IP address or range of IP addresses.

Preconfigured Access Policy	Description
	Note If you enabled VPC access for your domain, this preconfigured policy is not available. Instead, you can use security groups to control which IP addresses can access the domain. To learn more, see the section called "About Access Policies on VPC Domains" (p. 119).
Deny access to the domain	This policy allows access only through the Amazon ES console or by the owner of the AWS account who created the domain.
Copy access policy from another domain	This template provides a convenient way to import an existing access policy from another domain.
Allow open access to the domain	This policy is **not recommended** for domains with public endpoints. It allows anyone to delete, modify, or access indices in the domain. It is intended only as a convenience for testing. Don't load sensitive data into a domain that uses this setting.

6. Choose **Submit**.

Configuring Access Policies (AWS CLI)

Use the `--access-policies` option to configure access policies by using the AWS CLI. The following syntax is used by both the `create-elasticsearch-domain` and `update-elasticsearch-domain-config` commands.

Syntax

```
--access-policies=<value>
```

Parameter	Valid Values	Description
`--access-policies`	JSON	Specifies the access policy for the Amazon ES domain.

Example

The following resource-based policy example restricts access to the service to a single user, identified by the user's AWS account ID, 555555555555, in the `Principal` policy element. This user receives access to `index1`, but can't access other indices in the domain:

```
aws es update-elasticsearch-domain-config --domain-name mylogs --access-policies
 '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow","Principal": {"AWS":
 "arn:aws:iam::123456789012:root" },"Action":"es:*","Resource":"arn:aws:es:us-
east-1:555555555555:domain/index1/*" } ] }'
```

Tip
If you configure access policies using the AWS CLI, you can use one of many online tools to minify the JSON policy statement.

Configuring Access Policies (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon ES Configuration API Reference (p. 167), including the `--access-policies` parameter for `UpdateElasticsearchDomainConfig`. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring Automatic Snapshots

Amazon Elasticsearch Service provides automatic daily snapshots of a domain's primary index shards and the number of replica shards. By default, the service takes automatic snapshots at midnight, but you should choose a time when the service is under minimal load.

For information on working with these snapshots, see the section called “Restoring Snapshots” (p. 92).

> **Warning**
> The service stops taking snapshots of Amazon ES indices while the health of a cluster is red. Any documents that you add to a red cluster, even to indices with a health status of green, can be lost in the event of a cluster failure due to this lack of backups. To prevent loss of data, return the health of your cluster to green before uploading additional data to any index in the cluster. To learn more, see the section called “Red Cluster Status” (p. 143).

Configuring Snapshots (Console)

Use the following procedure to configure daily automatic index snapshots by using the console.

To configure automatic snapshots (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to update.
4. Choose **Configure cluster**.
5. For **Automated snapshot start hour**, choose the new hour for the service to take automated snapshots.
6. Choose **Submit**.

Configuring Snapshots (AWS CLI)

Use the following syntax for the `--snapshot-options` option. The syntax for the option is the same for both the `create-elasticsearch-domain` and `update-elasticsearch-domain-config` commands.

Syntax

```
--snapshot-options AutomatedSnapshotStartHour=<value>
```

Parameter	Valid Values	Description
`AutomatedSnapshotStartHour`	Integer between 0 and 23	Specifies the hour in UTC during which the service performs a daily automated snapshot of the indices in the new domain. The default

Parameter	Valid Values	Description
		value is 0, or midnight, which means that the snapshot is taken anytime between midnight and 1:00 AM.

Example

The following example configures automatic snapshots at 01:00 UTC:

```
aws es update-elasticsearch-domain-config --domain-name mylogs --region us-east-2 --
snapshot-options AutomatedSnapshotStartHour=1
```

Configuring Snapshots (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions that are defined in the Amazon ES Configuration API Reference (p. 167). This includes the `--snapshots-options` parameter for `UpdateElasticsearchDomainConfig`. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring Advanced Options

Use advanced options to configure the following:

rest.action.multi.allow_explicit_index

Specifies whether explicit references to indices are allowed inside the body of HTTP requests. Setting this property to false prevents users from bypassing access control for subresources. By default, the value is true. For more information, see the section called “Advanced Options and API Considerations” (p. 39).

indices.fielddata.cache.size

Specifies the percentage of Java heap space that is allocated to field data. By default, this setting is unbounded.

> **Note**
> Many customers query rotating daily indices. We recommend that you begin benchmark testing with `indices.fielddata.cache.size` configured to 40% of the JVM heap for most such use cases. However, if you have very large indices you might need a large field data cache.

indices.query.bool.max_clause_count

Specifies the maximum number of clauses allowed in a Lucene Boolean query. 1024 is the default. Queries with more than the permitted number of clauses result in a `TooManyClauses` error. For more information, see the Lucene documentation.

Configuring Advanced Options (Console)

To configure advanced options (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.

3. In the navigation pane, under **My domains**, choose the domain that you want to update.
4. Choose **Configure cluster**.
5. Choose **Advanced options**.
6. Specify the options that you want and choose **Submit**.

Configuring Advanced Options (AWS CLI)

Use the following syntax for the `--advanced-options` option. The syntax for the option is the same for both the `create-elasticsearch-domain` and `update-elasticsearch-domain-config` commands.

Syntax

```
--advanced-options rest.action.multi.allow_explicit_index=<true|false>,
 indices.fielddata.cache.size=<percentage_heap>, indices.query.bool.max_clause_count=<int>
```

Parameter	Valid Values
`--advanced-options`	`rest.action.multi.allow_explicit_index=<true|false>`
	`indices.fielddata.cache.size=<percentage_heap>`
	`indices.query.bool.max_clause_count=<int>`

Example

The following example disables explicit references to indices in the HTTP request bodies. It also limits the field data cache to 40 percent of the total Java heap:

```
aws es update-elasticsearch-domain-config --domain-name mylogs --region
 us-east-1 --advanced-options rest.action.multi.allow_explicit_index=false,
 indices.fielddata.cache.size=40
```

Configuring Advanced Options (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all of the actions defined in the Amazon ES Configuration API Reference (p. 167), including the `--advanced-options` parameter for `UpdateElasticsearchDomainConfig`. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring Logs

Amazon ES exposes three Elasticsearch logs through Amazon CloudWatch Logs: error logs, search slow logs, and index slow logs. These logs are useful for troubleshooting performance and stability issues, but are *disabled* by default. If enabled, standard CloudWatch pricing applies.

> **Note**
> Error logs are available only for Elasticsearch versions 5.1 and greater. Slow logs are available for all Elasticsearch versions.

For its logs, Elasticsearch uses Apache Log4j 2 and its built-in log levels (from least to most severe) of `TRACE`, `DEBUG`, `INFO`, `WARN`, `ERROR`, and `FATAL`. If you enable error logs, Amazon ES publishes log lines of `WARN`, `ERROR`, and `FATAL` to CloudWatch. Less severe levels are not available at this time. Error logs can help with troubleshooting in many situations, including:

- Painless script compilation issues
- Invalid queries
- Snapshot failures

Enabling Log Publishing (Console)

The Amazon ES console is the simplest way to enable the publishing of logs to CloudWatch.

To enable log publishing to CloudWatch (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to update.
4. On the **Logs** tab, choose **Enable** for the log that you want.
5. Create a CloudWatch log group, or choose an existing one.

   Note
   If you plan to enable multiple logs, we recommend publishing each to its own log group. This separation makes the logs easier to scan.
6. Choose an access policy that contains the appropriate permissions, or create a policy using the JSON that the console provides:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "cw_log_group_arn"
    }
  ]
}
```

   Important
   CloudWatch Logs supports 10 resource policies per region. If you plan to enable logs for several Amazon ES domains, you should create and reuse a broader policy that includes multiple log groups to avoid reaching this limit.
7. Choose **Enable**.

   The status of your domain changes from **Active** to **Processing**. The status must return to **Active** before log publishing is enabled. This process can take up to 30 minutes.

If you enabled one of the slow logs, see the section called “Setting Elasticsearch Logging Thresholds for Slow Logs” (p. 28). If you enabled only error logs, you don't need to perform any additional configuration steps.

Enabling Log Publishing (AWS CLI)

Before you can enable log publishing, you need a CloudWatch log group. If you don't already have one, you can create one using the following command:

```
aws logs create-log-group --log-group-name my-log-group
```

Type the next command to find the log group's ARN, and then *make a note of it*:

```
aws logs describe-log-groups --log-group-name my-log-group
```

Now you can give Amazon ES permissions to write to the log group. You must provide the log group's ARN near the end of the command:

```
aws logs put-resource-policy --policy-name my-policy --policy-document
 '{ "Version": "2012-10-17", "Statement": [{ "Sid": "", "Effect": "Allow",
 "Principal": { "Service": "es.amazonaws.com"}, "Action":[ "logs:PutLogEvents","
 logs:PutLogEventsBatch","logs:CreateLogStream"],"Resource": "cw_log_group_arn"}]}'
```

Important
CloudWatch Logs supports 10 resource policies per region. If you plan to enable slow logs for several Amazon ES domains, you should create and reuse a broader policy that includes multiple log groups to avoid reaching this limit.

Finally, you can use the `--log-publishing-options` option to enable publishing. The syntax for the option is the same for both the `create-elasticsearch-domain` and `update-elasticsearch-domain-config` commands.

Parameter	Valid Values
`--log-publishing-options`	`SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn=cw_log_group_arn,Enab false}`
	`INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn=cw_log_group_arn,Enabl false}`
	`ES_APPLICATION_LOGS={CloudWatchLogsLogGroupArn=cw_log_group_arn,E false}`

Note
If you plan to enable multiple logs, we recommend publishing each to its own log group. This separation makes the logs easier to scan.

Example

The following example enables the publishing of search and index slow logs for the specified domain:

```
aws es update-elasticsearch-domain-config --domain-name my-domain --log-publishing-options
 "SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-
group:my-log-
group,Enabled=true},INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-
east-1:123456789012:log-group:my-other-log-group,Enabled=true}"
```

To disable publishing to CloudWatch, run the same command with `Enabled=false`.

If you enabled one of the slow logs, see the section called “Setting Elasticsearch Logging Thresholds for Slow Logs” (p. 28). If you enabled only error logs, you don't need to perform any additional configuration steps.

Enabling Log Publishing (AWS SDKs)

Before you can enable log publishing, you must first create a CloudWatch log group, get its ARN, and give Amazon ES permissions to write to it. The relevant operations are documented in the Amazon CloudWatch Logs API Reference:

- `CreateLogGroup`
- `DescribeLogGroup`
- `PutResourcePolicy`

You can access these operations using the AWS SDKs.

The AWS SDKs (except the Android and iOS SDKs) support all the operations that are defined in the Amazon ES Configuration API Reference (p. 167), including the `--log-publishing-options` option for `CreateElasticsearchDomain` and `UpdateElasticsearchDomainConfig`.

If you enabled one of the slow logs, see the section called “Setting Elasticsearch Logging Thresholds for Slow Logs” (p. 28). If you enabled only error logs, you don't need to perform any additional configuration steps.

Setting Elasticsearch Logging Thresholds for Slow Logs

Elasticsearch disables slow logs by default. After you enable the *publishing* of slow logs to CloudWatch, you still must specify logging thresholds for each Elasticsearch index. These thresholds define precisely what should be logged and at which log level. Settings vary slightly by Elasticsearch version.

You specify these settings through the Elasticsearch REST API:

```
PUT elasticsearch_domain_endpoint/index/_settings
{
  "index.search.slowlog.threshold.query.warn": "5s",
  "index.search.slowlog.threshold.query.info": "2s"
}
```

To test that slow logs are publishing successfully, consider starting with very low values to verify that logs appear in CloudWatch, and then increase the thresholds to more useful levels.

If the logs don't appear, check the following:

- Does the CloudWatch log group exist? Check the CloudWatch console.
- Does Amazon ES have permissions to write to the log group? Check the Amazon ES console.
- Is the Amazon ES domain configured to publish to the log group? Check the Amazon ES console, use the AWS CLI `describe-elasticsearch-domain-config` option, or call `DescribeElasticsearchDomainConfig` using one of the SDKs.
- Are the Elasticsearch logging thresholds low enough that your requests are exceeding them? To review your thresholds for an index, use the following command:

  ```
  GET elasticsearch_domain_endpoint/index/_settings?pretty
  ```

If you want to disable slow logs for an index, return any thresholds that you changed to their default values of `-1`.

Disabling publishing to CloudWatch using the Amazon ES console or AWS CLI does *not* stop Elasticsearch from generating logs; it only stops the *publishing* of those logs. Be sure to check your index settings if you no longer need the slow logs.

Viewing Logs

Viewing the application and slow logs in CloudWatch is just like viewing any other CloudWatch log. For more information, see View Log Data in the *Amazon CloudWatch Logs User Guide*.

Here are some considerations for viewing the logs:

- Amazon ES publishes only the first 255,000 characters of each line to CloudWatch. Any remaining content is truncated.
- In CloudWatch, the log stream names have suffixes of `-index-slow-logs`, `-search-slow-logs`, and `-es-application-logs` to help identify their contents.

Amazon Elasticsearch Service Access Control

Amazon Elasticsearch Service offers several ways of controlling access to your domains. This section covers the various policy types, how they interact with each other, and how to create your own, custom policies.

> **Important**
> VPC support introduces some additional considerations to Amazon ES access control. For more information, see the section called “About Access Policies on VPC Domains” (p. 119).

Types of Policies

Amazon ES supports three types of access policies:

- the section called “Resource-based Policies” (p. 30)
- the section called “Identity-based policies” (p. 32)
- the section called “IP-based Policies” (p. 33)

Resource-based Policies

You attach resource-based policies to domains. These policies specify which actions a principal can perform on the domain's *subresources*. Subresources include Elasticsearch indices and APIs.

The `Principal` element specifies the accounts, users, or roles that are allowed access. The `Resource` element specifies which subresources these principals can access. The following resource-based policy grants `test-user` full access (`es:*`) to `test-domain`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:*"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

Two important considerations apply to this policy:

- These privileges apply only to this domain. Unless you create additional policies, `test-user` can't access other domains or even view a list of them in the Amazon ES dashboard.
- The trailing `/*` in the `Resource` element is significant. Despite having full access, `test-user` can perform these actions only on the domain's subresources, not on the domain's configuration.

For example, test-user can make requests against an index (GET https://search-test-domain.us-west-1.es.amazonaws.com/test-index), but can't update the domain's configuration (POST https://es.us-west-1.amazonaws.com/2015-01-01/es/domain/test-domain/config). Note the difference between the two endpoints. Accessing the configuration API (p. 167) requires an identity-based policy (p. 32).

To further restrict test-user, you can apply the following policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/test-index/_search"
    }
  ]
}
```

Now test-user can perform only one operation: searches against test-index. All other indices within the domain are inaccessible, and without permissions to use the es:ESHttpPut or es:ESHttpPost actions, test-user can't add or modify documents.

Next, you might decide to configure a role for power users. This policy allows power-user-role access to all HTTP methods, except for the ability to delete a critical index and its documents:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/power-user-role"
        ]
      },
      "Action": [
        "es:ESHttpDelete",
        "es:ESHttpGet",
        "es:ESHttpHead",
        "es:ESHttpPost",
        "es:ESHttpPut"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    },
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/power-user-role"
        ]
      },
      "Action": [
        "es:ESHttpDelete"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/critical-index*"
    }
  ]
}
```

For information about all available actions, see the section called “Policy Element Reference” (p. 35).

Identity-based policies

Unlike resource-based policies, which you attach to domains in Amazon ES, you attach identity-based policies to users or roles using the AWS Identity and Access Management (IAM) service. Just like resource-based policies (p. 30), identity-based policies specify who can access a service, which actions they can perform, and if applicable, the resources on which they can perform those actions.

While they certainly don't have to be, identity-based policies tend to be more generic. They often govern the basic, service-level actions a user can perform. After you have these policies in place, you can use resource-based policies in Amazon ES to offer users additional permissions.

Because identity-based policies attach to users or roles (principals), the JSON doesn't specify a principal. The following policy grants access to actions that begin with `Describe` and `List` and allows `GET` requests against all domains. This combination of actions provides read-only access:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "es:Describe*",
        "es:List*",
        "es:ESHttpGet"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

An administrator might have full access to Amazon ES:

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

For more information about the differences between resource-based and identity-based policies, see IAM Policies in the *IAM User Guide*.

> **Note**
> Users with the AWS managed `AmazonESReadOnlyAccess` policy can't see cluster health status in the console. To allow them to see cluster health status, add the `"es:ESHttpGet"` action to an access policy and attach it to their accounts or roles.

IP-based Policies

IP-based policies restrict access to a domain to one or more IP addresses or CIDR blocks. Technically, IP-based policies are not a distinct type of policy. Instead, they are just resource-based policies that specify an anonymous principal and include a special `Condition` element.

The primary appeal of IP-based policies is that they allow unsigned requests to an Amazon ES domain, which lets you use clients like curl and the section called “Kibana” (p. 99) or access the domain through a proxy server. To learn more, see the section called “Using a Proxy to Access Amazon ES from Kibana” (p. 99).

> **Note**
> If you enabled VPC access for your domain, you can't configure an IP-based policy. Instead, you can use security groups to control which IP addresses can access the domain. For more information, see the section called “About Access Policies on VPC Domains” (p. 119).

The following IP-based access policy grants all requests that originate from `12.345.678.901` access to `test-domain`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "12.345.678.901"
          ]
        }
      },
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

Making and Signing Amazon ES Requests

Even if you configure a completely open resource-based access policy, *all* requests to the Amazon ES configuration API must be signed. If your policies specify IAM users or roles, requests to the Elasticsearch APIs also must be signed. The signing method differs by API:

- To make calls to the Amazon ES configuration API, we recommend that you use one of the AWS SDKs. The SDKs greatly simplify the process and can save you a significant amount of time compared to creating and signing your own requests. The configuration API endpoints use the following format:

  ```
  es.region.amazonaws.com/2015-01-01/
  ```

  For example, the following request makes a minor configuration change to the `movies` domain, but you have to sign it yourself (not recommended):

  ```
  POST https://es.us-east-1.amazonaws.com/2015-01-01/es/domain/movies/config
  ```

```
{
  "SnapshotOptions": {
    "AutomatedSnapshotStartHour": 3
  }
}
```

If you use one of the SDKs, such as Boto 3, the SDK automatically handles the request signing:

```
import boto3

client = boto3.client('es')
response = client.update_elasticsearch_domain_config(
  DomainName='movies',
  SnapshotOptions={
    'AutomatedSnapshotStartHour': 3
  }
)
```

- To make calls to the Elasticsearch APIs, you must sign your own requests. For sample code, see *Signing HTTP Requests* (p. 62). The Elasticsearch APIs use the following format:

  ```
  domain.region.es.amazonaws.com
  ```

  For example, the following request searches the `movies` index for *thor*:

  ```
  GET https://search-my-domain.us-east-1.es.amazonaws.com/movies/_search?q=thor
  ```

Amazon ES supports authentication using AWS Signature Version 4. For more information, see Signature Version 4 Signing Process.

> **Note**
> The service ignores parameters passed in URLs for HTTP POST requests that are signed with Signature Version 4.

When Policies Collide

Complexities arise when policies disagree or make no explicit mention of a user. Understanding How IAM Works in the *IAM User Guide* provides a concise summary of policy evaluation logic:

- By default, all requests are denied.
- An explicit allow overrides this default.
- An explicit deny overrides any allows.

For example, if a resource-based policy grants you access to a domain, but an identify-based policy denies you access, you are denied access. If an identity-based policy grants access and a resource-based policy does not specify whether or not you should have access, you are allowed access. See the following table of intersecting policies for a full summary of outcomes.

	Allowed in Resource-based Policy	**Denied in Resource-based Policy**	**Neither Allowed nor Denied in Resource-based Policy**
Allowed in Identity-based Policy	Allow	Deny	Allow

	Allowed in Resource-based Policy	**Denied in Resource-based Policy**	**Neither Allowed nor Denied in Resource-based Policy**
Denied in Identity-based Policy	Deny	Deny	Deny
Neither Allowed nor Denied in Identity-based Policy	Allow	Deny	Deny

Policy Element Reference

Amazon ES supports most policy elements in the IAM Policy Elements Reference, with the exception of `NotPrincipal`. The following table shows the most common elements.

JSON Policy Element	Summary
`Version`	The current version of the policy language is `2012-10-17`. All access policies should specify this value.
`Effect`	This element specifies whether the statement allows or denies access to the specified actions. Valid values are `Allow` or `Deny`.
`Principal`	This element specifies the AWS account or IAM user or role that is allowed or denied access to a resource and can take several forms: • **AWS accounts**: `"Principal":{"AWS": ["123456789012"]}` or `"Principal":{"AWS": ["arn:aws:iam::123456789012:root"]}` • **IAM users**: `"Principal":{"AWS": ["arn:aws:iam::123456789012:user/test-user"]}` • **IAM roles**: `"Principal":{"AWS": ["arn:aws:iam::123456789012:role/test-role"]}` Specifying the `*` wildcard enables anonymous access to the domain, which we don't recommend unless you add an IP-based condition.
`Action`	Amazon ES uses the following actions for HTTP methods: • `es:ESHttpDelete` • `es:ESHttpGet` • `es:ESHttpHead` • `es:ESHttpPost` • `es:ESHttpPut` Amazon ES uses the following actions for the configuration API (p. 167): • `es:AddTags` • `es:CreateElasticsearchDomain` • `es:DeleteElasticsearchDomain` • `es:DeleteElasticsearchServiceRole` • `es:DescribeElasticsearchDomain`

JSON Policy Element	Summary

- `es:DescribeElasticsearchDomainConfig`
- `es:DescribeElasticsearchDomains`
- `es:DescribeElasticsearchInstanceTypeLimits`
- `es:ListDomainNames`
- `es:ListElasticsearchInstanceTypeDetails`
- `es:ListElasticsearchInstanceTypes`
- `es:ListElasticsearchVersions`
- `es:ListTags`
- `es:RemoveTags`
- `es:UpdateElasticsearchDomainConfig`

> **Tip**
> You can use wildcards to specify a subset of actions, such as `"Action":"es:*"` or `"Action":"es:Describe*"`.

Certain `es:` actions support resource-level permissions. For example, you can give a user permissions to delete one particular domain without giving that user permissions to delete *any* domain. Other actions apply only to the service itself. For example, `es:ListDomainNames` makes no sense in the context of a single domain and thus requires a wildcard.

> **Important**
> Resource-based policies differ from resource-level permissions. Resource-based policies (p. 30) are full JSON policies that attach to domains. Resource-level permissions let you restrict actions to particular domains or subresources. In practice, you can think of resource-level permissions as an optional part of a resource- or identity-based policy.

The following identity-based policy (p. 32) lists all `es:` actions and groups them according to whether they apply to the domain subresources (`test-domain/*`), to the domain configuration (`test-domain`), or only to the service (`*`):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpDelete",
        "es:ESHttpGet",
        "es:ESHttpHead",
        "es:ESHttpPost",
        "es:ESHttpPut"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-
domain/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "es:CreateElasticsearchDomain",
        "es:DeleteElasticsearchDomain",
        "es:DescribeElasticsearchDomain",
```

<table>
<tr><th>JSON Policy Element</th><th>Summary</th></tr>
<tr><td></td><td>

```
        "es:DescribeElasticsearchDomainConfig",
        "es:DescribeElasticsearchDomains",
        "es:UpdateElasticsearchDomainConfig"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain"
    },
    {
      "Effect": "Allow",
      "Action": [
        "es:AddTags",
        "es:DeleteElasticsearchServiceRole",
        "es:DescribeElasticsearchInstanceTypeLimits",
        "es:ListDomainNames",
        "es:ListElasticsearchInstanceTypeDetails",
        "es:ListElasticsearchInstanceTypes",
        "es:ListElasticsearchVersions",
        "es:ListTags",
        "es:RemoveTags"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

While resource-level permissions for `es:CreateElasticsearchDomain` might seem unintuitive—after all, why give a user permissions to create a domain that already exists?—the use of a wildcard lets you enforce a simple naming scheme for your domains, such as `"Resource": "arn:aws:es:us-west-1:987654321098:domain/my-team-name-*"`.

Of course, nothing prevents you from including actions alongside less restrictive resource elements, such as the following:

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

To learn more about pairing actions and resources, see the `Resource` element in this table.

</td></tr>
</table>

<table>
<tr><th>JSON Policy Element</th><th>Summary</th></tr>
<tr><td><code>Condition</code></td><td>Amazon ES supports most conditions that are described in Available Global Condition Keys in the <em>IAM User Guide</em>. One notable exception is the <code>aws:SecureTransport</code> key, which Amazon ES does not support.
When configuring an IP-based policy (p. 33), you specify the IP addresses or CIDR block as a condition, such as the following:<pre><code>"Condition": {
  "IpAddress": {
    "aws:SourceIp": [
      "192.0.2.0/32"
    ]
  }
}</code></pre></td></tr>
</table>

JSON Policy Element	Summary

`Resource`	Amazon ES uses `Resource` elements in three basic ways:
	• For actions that apply to Amazon ES itself, like `es:ListDomainNames`, or to allow full access, use the following syntax:
	`"Resource": "*"`
	• For actions that involve a domain's configuration, like `es:DescribeElasticsearchDomain`, you can use the following syntax:
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name"`
	• For actions that apply to a domain's subresources, like `es:ESHttpGet`, you can use the following syntax:
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name/*"`
	You don't have to use a wildcard. Amazon ES lets you define a different access policy for each Elasticsearch index or API. For example, you might limit a user's permissions to the `test-index` index:
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name/test-index"`
	Instead of full access to `test-index`, you might prefer to limit the policy to just the search API.
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name/test-index/_search"`
	You can even control access to individual documents:
	`"Resource": "arn:aws:es:region:aws-account-id:domain/domain-name/test-index/test-type/1"`
	Essentially, if Elasticsearch expresses the subresource as an endpoint, you can control access to it.
	For details about which actions support resource-level permissions, see the `Action` element in this table.

Advanced Options and API Considerations

Amazon ES has several advanced options, one of which has access control implications: `rest.action.multi.allow_explicit_index`. At its default setting of true, it allows users to bypass subresource permissions under certain circumstances.

For example, consider the following resource-based policy:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": [
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/test-index/*",
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/_bulk"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/restricted-index/*"
    }
  ]
}
```

This policy grants `test-user` full access to `test-index` and the Elasticsearch bulk API. It also allows `GET` requests to `restricted-index`.

The following indexing request, as you might expect, fails due to a permissions error:

```
PUT https://search-test-domain.us-west-1.es.amazonaws.com/restricted-index/movie/1
{
  "title": "Your Name",
  "director": "Makoto Shinkai",
  "year": "2016"
}
```

Unlike the index API, the bulk API lets you create, update, and delete many documents in a single call. You often specify these operations in the request body, however, rather than in the request URL. Because Amazon ES uses URLs to control access to domain subresources, `test-user` can, in fact, use the bulk API to make changes to `restricted-index`. Even though the user lacks `POST` permissions on the index, the following request **succeeds**:

```
POST https://search-test-domain.us-west-1.es.amazonaws.com/_bulk
{ "index" : { "_index": "restricted-index", "_type" : "movie", "_id" : "1" } }
{ "title": "Your Name", "director": "Makoto Shinkai", "year": "2016" }
```

In this situation, the access policy fails to fulfill its intent. To prevent users from bypassing these kinds of restrictions, you can change `rest.action.multi.allow_explicit_index` to false. If this value is false, all calls to the bulk, mget, and msearch APIs that specify index names in the request body stop working. In other words, calls to `_bulk` no longer work, but calls to `test-index/_bulk` do. This second endpoint contains an index name, so you don't need to specify one in the request body.

Kibana (p. 99) relies heavily on mget and msearch, so it is unlikely to work properly after this change. For partial remediation, you can leave `rest.action.multi.allow_explicit_index` as true and deny certain users access to one or more of these APIs.

For information about changing this setting, see the section called “Configuring Advanced Options” (p. 24).

Similarly, the following resource-based policy contains two subtle issues:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    },
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/restricted-index/*"
    }
  ]
}
```

- Despite the explicit deny, `test-user` can still make calls such as `GET https://search-test-domain.us-west-1.es.amazonaws.com/_all/_search` and `GET https://search-test-domain.us-west-1.es.amazonaws.com/*/_search` to access the documents in `restricted-index`.
- Because the `Resource` element references `restricted-index/*`, `test-user` doesn't have permissions to directly access the index's documents. The user does, however, have permissions to *delete the entire index*. To prevent access and deletion, the policy instead must specify `restricted-index*`.

Rather than mixing broad allows and focused denies, the safest approach is to follow the principle of least privilege and grant only the permissions that are required to perform a task.

Configuring Access Policies

- For instructions on creating or modifying resource- and IP-based policies in Amazon ES, see the section called “Configuring Access Policies” (p. 21).
- For instructions on creating or modifying identity-based policies in IAM, see Creating IAM Policies in the *IAM User Guide*.

Additional Sample Policies

Although this chapter includes many sample policies, AWS access control is a complex subject that is best understood through examples. For more, see Example Policies in the *IAM User Guide*.

Managing Amazon Elasticsearch Service Domains

As the size and number of documents in your Amazon Elasticsearch Service (Amazon ES) domain grow and as network traffic increases, you likely will need to update the configuration of your Elasticsearch cluster. To know when it's time to reconfigure your domain, you need to monitor domain metrics. You might also need to audit data-related API calls to your domain or assign tags to your domain. This section describes how to perform these and other tasks related to managing your domains.

Topics

About Configuration Changes

Amazon ES uses a *blue/green* deployment process when updating domains. Blue/green typically refers to the practice of running two production environments, one live and one idle, and switching the two as you make software changes. In the case of Amazon ES, it refers to the practice of creating a new environment for domain updates and routing users to the new environment after those updates are complete. The practice minimizes downtime and maintains the original environment in the event that deployment to the new environment is unsuccessful.

The following operations cause blue/green deployments:

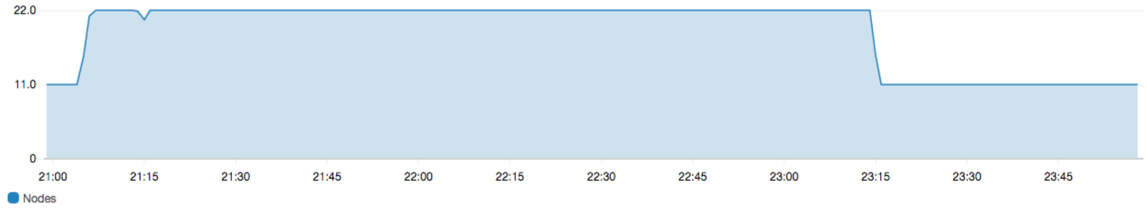
- Changing instance count or type
- Enabling or disabling dedicated master nodes
- Changing dedicated master node count
- Enabling or disabling zone awareness
- Changing storage type, volume type, or volume size
- Choosing different VPC subnets
- Adding or removing VPC security groups
- Enabling or disabling Amazon Cognito authentication for Kibana
- Choosing a different Amazon Cognito user pool or identity pool
- Modifying advanced settings
- Enabling or disabling the publication of error logs or slow logs to CloudWatch
- Upgrading to a new Elasticsearch version

The following operations do **not** cause blue/green deployments:

- Changing access policy

- Changing automated snapshot hour

Domain updates also occur when the Amazon ES team makes certain software changes to the service. If you initiate a configuration change, the domain state changes to **Processing**. If the Amazon ES team makes software changes, the state remains **Active**. In both cases, you can review the cluster health and Amazon CloudWatch metrics and see that the number of nodes in the cluster temporarily increases—often doubling—while the domain update occurs. In the following illustration, you can see the number of nodes doubling from 11 to 22 during a configuration change and returning to 11 when the update is complete.



This temporary increase can strain the cluster's dedicated master nodes (p. 128), which suddenly have many more nodes to manage. It is important to maintain sufficient capacity on dedicated master nodes to handle the overhead that is associated with these blue/green deployments.

> **Important**
> You do *not* incur any additional charges during configuration changes and service maintenance. You are billed only for the number of nodes that you request for your cluster. For specifics, see the section called "Charges for Configuration Changes" (p. 43).

To prevent overloading dedicated master nodes, you can monitor usage with the Amazon CloudWatch metrics (p. 46). For recommended maximum values, see the section called "Recommended CloudWatch Alarms" (p. 130).

Charges for Configuration Changes

If you change the configuration for a domain, Amazon ES creates a new cluster as described in the section called "About Configuration Changes" (p. 42). During the migration of old to new, you incur the following charges:

- If you change the instance type, you are charged for both clusters for the first hour. After the first hour, you are charged only for the new cluster.

  Example: You change the configuration from three `m3.xlarge` instances to four `m4.large` instances. For the first hour, you are charged for both clusters (3 * `m3.xlarge` + 4 * `m4.large`). After the first hour, you are charged only for the new cluster (4 * `m4.large`).
- If you don't change the instance type, you are charged only for the largest cluster for the first hour. After the first hour, you are charged only for the new cluster.

  Example: You change the configuration from six `m3.xlarge` instances to three `m3.xlarge` instances. For the first hour, you are charged for the largest cluster (6 * `m3.xlarge`). After the first hour, you are charged only for the new cluster (3 * `m3.xlarge`).

Enabling Zone Awareness

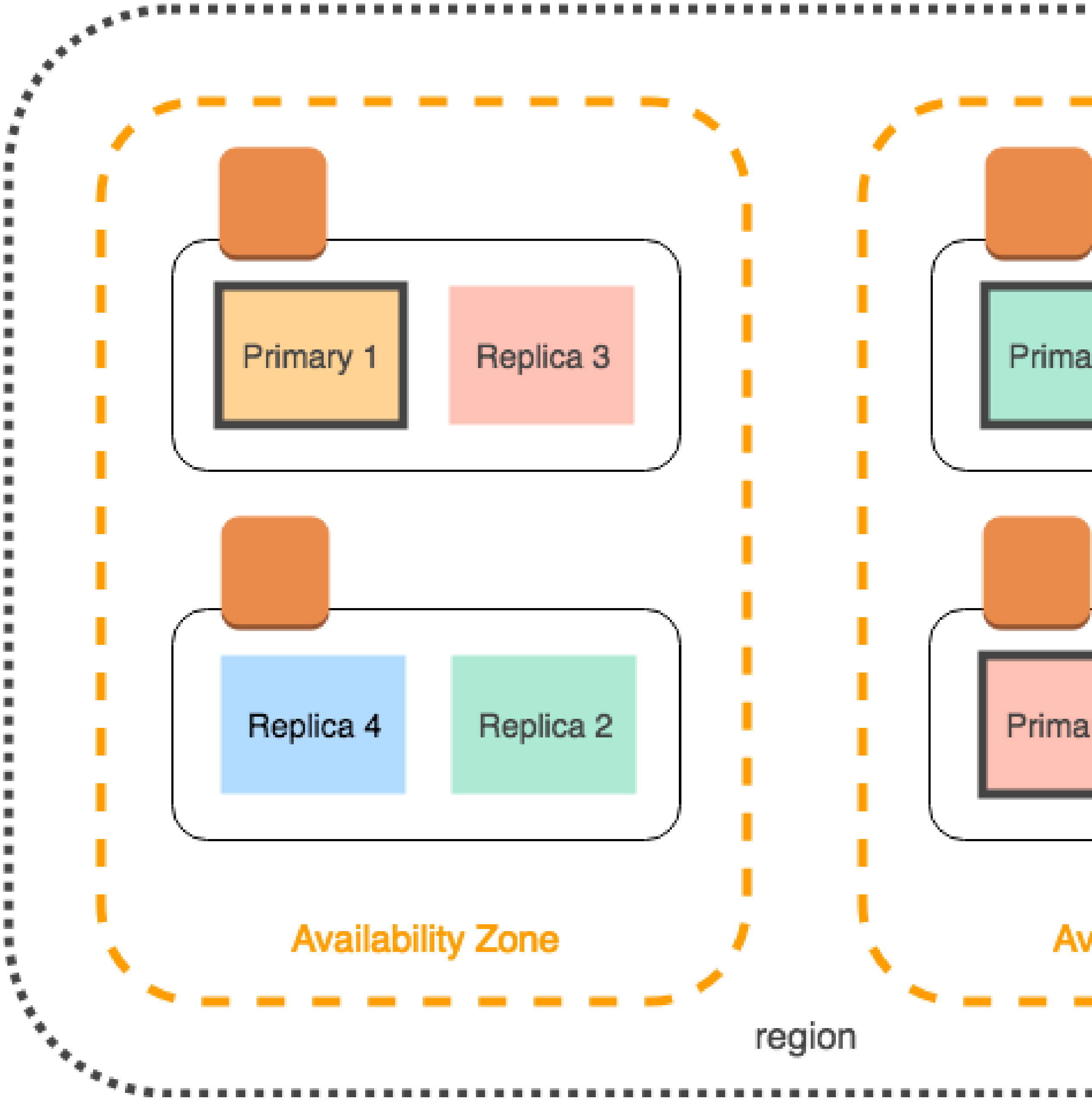
Each AWS Region is a separate geographic area with multiple, isolated locations known as *Availability Zones*. To prevent data loss and minimize downtime in the event of node and data center failure, you can use the Amazon ES console to allocate an Elasticsearch cluster's nodes and shards across two Availability Zones in the same region. This allocation is known as *zone awareness*. Zone awareness requires an even number of instances and slightly increases network latencies.

If you enable zone awareness, you must have at least one replica for each index in your cluster. Fortunately, the default configuration for any index is a replica count of 1. Amazon ES distributes primary and replica shards across nodes in different Availability Zones, which increases the availability of your cluster.

> **Important**
> If you specify a replica count of 0 for an index, enabling zone awareness doesn't provide any additional availability; without replicas, Amazon ES can't distribute copies of your data to other Availability Zones.

If you enable zone awareness and use VPC access domains, you must specify Availability Zones for the VPC subnets. For more information about VPCs, see *VPC Support* (p. 117).

The following illustration shows a four-node cluster with zone awareness enabled. The service distributes the shards so that no replica shard is in the same Availability Zone as its corresponding primary shard.



If one Availability Zone (AZ) experiences a service interruption, you have a 50/50 chance of cluster downtime due to how master node (p. 128) election works. For example, if you use the recommended three dedicated master nodes, Amazon ES distributes two dedicated master nodes into one AZ and one dedicated master node into the other. If the AZ with two dedicated master nodes experiences an interruption, your cluster is unavailable until the remaining AZ can automatically replace the now-missing dedicated master nodes, achieve a quorum, and elect a new master.

Further, if one AZ experiences an interruption, the cluster's data nodes might experience a period of extreme load while Amazon ES automatically configures new nodes to replace the now-missing ones. Suddenly, half as many nodes have to process just as many requests to the cluster. As they process these requests, the remaining nodes are also struggling to replicate data onto new nodes as they come online. A cluster with extra resources can alleviate this concern.

To enable zone awareness (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose your Amazon ES domain.
4. Choose **Configure cluster**.
5. In the **Node configuration** pane, choose **Enable zone awareness**.
6. Choose **Submit**.

For more information, see Regions and Availability Zones in the EC2 documentation.

Monitoring Cluster Metrics and Statistics with Amazon CloudWatch (Console)

Amazon ES domains send performance metrics to Amazon CloudWatch every minute. If you use general purpose or magnetic EBS volumes, the EBS volume metrics only update every five minutes. These metrics are available for all Amazon ES domains at no extra charge.

Amazon ES metrics fall into categories:

- the section called "Cluster Metrics" (p. 46)
- the section called "Dedicated Master Node Metrics" (p. 49)
- the section called "EBS Volume Metrics" (p. 50)
- the section called "Instance Metrics" (p. 51)

> **Note**
> The service archives the metrics for two weeks before discarding them.

To view configurable statistics for a metric (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose your Amazon ES domain.
4. Choose the **Monitoring** tab.
5. Choose the metric that you want to view.
6. From the **Statistic** list, select a statistic.

   For a list of relevant statistics for each metric, see the tables in Cluster Metrics (p. 46). Some statistics are not relevant for a given metric. For example, the **Sum** statistic is not meaningful for the **Nodes** metric.
7. Choose **Update graph**.

Cluster Metrics

The `AWS/ES` namespace includes the following metrics for clusters.

Metric	Description
`ClusterStatus.green`	Indicates that all index shards are allocated to nodes in the cluster. Relevant statistics: Minimum, Maximum
`ClusterStatus.yellow`	Indicates that the primary shards for all indices are allocated to nodes in a cluster, but the replica shards for at least one index are not. Single node clusters always initialize with this cluster status because there is no second node to which a replica can be assigned. You can either increase your node count to obtain a green cluster status, or you can use the Elasticsearch API to set the `number_of_replicas` setting for your index to `0`. To learn more, see Configuring Amazon Elasticsearch Service Domains. Relevant statistics: Minimum, Maximum
`ClusterStatus.red`	Indicates that the primary and replica shards of at least one index are not allocated to nodes in a cluster. To recover, you must delete the indices or restore a snapshot and then add EBS-based storage, use larger instance types, or add instances. For more information, see Red Cluster Status. Relevant statistics: Minimum, Maximum
`Nodes`	The number of nodes in the Amazon ES cluster, including dedicated master nodes. Relevant statistics: Minimum, Maximum, Average
`SearchableDocuments`	The total number of searchable documents across all indices in the cluster. Relevant statistics: Minimum, Maximum, Average
`DeletedDocuments`	The total number of documents marked for deletion across all indices in the cluster. These documents no longer appear in search results, but Elasticsearch only removes deleted documents from disk during segment merges. This metric increases after delete requests and decreases after segment merges. Relevant statistics: Minimum, Maximum, Average
`CPUUtilization`	The maximum percentage of CPU resources used for data nodes in the cluster. Relevant statistics: Maximum, Average
`FreeStorageSpace`	The free space, in megabytes, for nodes in the cluster. `Sum` shows total free space for the cluster, but you must leave the period at one minute to get an accurate count. `Minimum`, `Maximum`, and `Average` show free space for individual nodes. Amazon ES throws a `ClusterBlockException` when this metric reaches `0`. To recover, you must either delete indices, add larger instances, or add EBS-based storage to existing instances. To learn more, see Recovering from a Lack of Free Storage Space **Note** `FreeStorageSpace` will always be lower than the value that the Elasticsearch `_cluster/stats` API provides. Amazon ES

Metric	Description
	reserves a percentage of the storage space on each instance for internal operations. Relevant statistics: Minimum, Maximum, Average, Sum
`ClusterUsedSpace`	The total used space, in megabytes, for a cluster. You can view this metric in the Amazon CloudWatch console, but not in the Amazon ES console. Relevant statistics: Minimum, Maximum
`ClusterIndexWritesBlocked`	Indicates whether your cluster is accepting or blocking incoming write requests. A value of 0 means that the cluster is accepting requests. A value of 1 means that it is blocking requests. Many factors can cause a cluster to begin blocking requests. Some common factors include the following: `FreeStorageSpace` is too low, `JVMMemoryPressure` is too high, or `CPUUtilization` is too high. To alleviate this issue, consider adding more disk space or scaling your cluster. Relevant statistics: Maximum **Note** You can view this metric in the Amazon CloudWatch console, but not the Amazon ES console.
`JVMMemoryPressure`	The maximum percentage of the Java heap used for all data nodes in the cluster. Relevant statistics: Maximum
`AutomatedSnapshotFailure`	The number of failed automated snapshots for the cluster. A value of `1` indicates that no automated snapshot was taken for the domain in the previous 36 hours. Relevant statistics: Minimum, Maximum
`CPUCreditBalance`	The remaining CPU credits available for data nodes in the cluster. A CPU credit provides the performance of a full CPU core for one minute. For more information, see CPU Credits in the *Amazon EC2 Developer Guide*. This metric is available only for the t2.micro.elasticsearch, t2.small.elasticsearch, and t2.medium.elasticsearch instance types. Relevant statistics: Minimum
`KibanaHealthyNodes`	A health check for Kibana. A value of 1 indicates normal behavior. A value of 0 indicates that Kibana is inaccessible. In most cases, the health of Kibana mirrors the health of the cluster. Relevant statistics: Minimum **Note** You can view this metric on the Amazon CloudWatch console, but not the Amazon ES console.

Metric	Description
`KMSKeyError`	A value of 1 indicates that the KMS customer master key used to encrypt data at rest has been disabled. To restore the domain to normal operations, re-enable the key. The console displays this metric only for domains that encrypt data at rest. Relevant statistics: Minimum, Maximum
`KMSKeyInaccessible`	A value of 1 indicates that the KMS customer master key used to encrypt data at rest has been deleted or revoked its grants to Amazon ES. You can't recover domains that are in this state. If you have a manual snapshot, though, you can use it to migrate the domain's data to a new domain. The console displays this metric only for domains that encrypt data at rest. Relevant statistics: Minimum, Maximum
`InvalidHostHeaderRequests`	The number of HTTP requests made to the Elasticsearch cluster that included an invalid (or missing) host header. Valid requests include the domain endpoint as the host header value. If you see large values for this metric, check that your Elasticsearch clients include the proper host header value in their requests. Otherwise, Amazon ES might reject the requests. You can also update the domain’s access policy to require signed requests. Relevant statistics: Sum
`ElasticsearchRequests`	The number of requests made to the Elasticsearch cluster. Relevant statistics: Sum
`RequestCount`	The number of requests to a domain and the HTTP response code (2xx, 3xx, 4xx, 5xx) for each request. Relevant statistics: Sum

Dedicated Master Node Metrics

The `AWS/ES` namespace includes the following metrics for dedicated master nodes.

Metric	Description
`MasterCPUUtilization`	The maximum percentage of CPU resources used by the dedicated master nodes. We recommend increasing the size of the instance type when this metric reaches 60 percent. Relevant statistics: Average
`MasterFreeStorageSpace`	This metric is not relevant and can be ignored. The service does not use master nodes as data nodes.
`MasterJVMMemoryPressure`	The maximum percentage of the Java heap used for all dedicated master nodes in the cluster. We recommend moving to a larger instance type when this metric reaches 85 percent. Relevant statistics: Maximum

Metric	Description
`MasterCPUCreditBalance`	The remaining CPU credits available for dedicated master nodes in the cluster. A CPU credit provides the performance of a full CPU core for one minute. For more information, see CPU Credits in the *Amazon EC2 User Guide for Linux Instances*. This metric is available only for the t2.micro.elasticsearch, t2.small.elasticsearch, and t2.medium.elasticsearch instance types. Relevant statistics: Minimum
`MasterReachableFromNode`	A health check for `MasterNotDiscovered` exceptions. A value of 1 indicates normal behavior. A value of 0 indicates that `/_cluster/health/` is failing. Failures mean that the master node stopped or is not reachable. They are usually the result of a network connectivity issue or AWS dependency problem. Relevant statistics: Minimum **Note** You can view this metric on the Amazon CloudWatch console, but not the Amazon ES console.

EBS Volume Metrics

The `AWS/ES` namespace includes the following metrics for EBS volumes.

Metric	Description
`ReadLatency`	The latency, in seconds, for read operations on EBS volumes. Relevant statistics: Minimum, Maximum, Average
`WriteLatency`	The latency, in seconds, for write operations on EBS volumes. Relevant statistics: Minimum, Maximum, Average
`ReadThroughput`	The throughput, in bytes per second, for read operations on EBS volumes. Relevant statistics: Minimum, Maximum, Average
`WriteThroughput`	The throughput, in bytes per second, for write operations on EBS volumes. Relevant statistics: Minimum, Maximum, Average
`DiskQueueDepth`	The number of pending input and output (I/O) requests for an EBS volume. Relevant statistics: Minimum, Maximum, Average
`ReadIOPS`	The number of input and output (I/O) operations per second for read operations on EBS volumes. Relevant statistics: Minimum, Maximum, Average
`WriteIOPS`	The number of input and output (I/O) operations per second for write operations on EBS volumes.

Metric	Description
	Relevant statistics: Minimum, Maximum, Average

Instance Metrics

The `AWS/ES` namespace includes the following metrics for each instance in a domain.

Metric	Description
`IndexingLatency`	The average time, in milliseconds, to process an indexing request. Relevant statistics: Average
`IndexingRate`	The number of indexing operations per minute. A single call to the `_bulk` API that adds two documents, updates two, and deletes two qualifies as six indexing operations. Relevant statistics: Average
`SearchLatency`	The average time, in milliseconds, to process a search request. Relevant statistics: Average
`SearchRate`	The number of search requests per minute. Relevant statistics: Average
`SysMemoryUtilization`	The percentage of the instance's memory that is in use. Relevant statistics: Minimum, Maximum, Average
`JVMGCYoungCollectionCount`	The number of times that "young generation" garbage collection has run. A large number of runs is a normal part of cluster operations. Relevant statistics: Maximum
`JVMGCYoungCollectionTime`	The amount of time, in milliseconds, that the cluster has spent performing "young generation" garbage collection. Relevant statistics: Maximum
`JVMGCOldCollectionCount`	The number of times that "old generation" garbage collection has run. In a cluster with sufficient resources, this number should remain small. Relevant statistics: Maximum
`JVMGCOldCollectionTime`	The amount of time, in milliseconds, that the cluster has spent performing "old generation" garbage collection. Relevant statistics: Maximum
`ThreadpoolForce_mergeQueue`	The number of queued tasks in the force merge thread pool. If the queue size is consistently high, consider scaling your cluster. Relevant statistics: Maximum
`ThreadpoolForce_mergeRejected`	The number of rejected tasks in the force merge thread pool. If this number continually grows, consider scaling your cluster.

Metric	Description
	Relevant statistics: Maximum
`ThreadpoolForce_mergeThreads`	The size of the force merge thread pool. Relevant statistics: Maximum
`ThreadpoolIndexQueue`	The number of queued tasks in the index thread pool. If the queue size is consistently high, consider scaling your cluster. The maximum index queue size is 200. Relevant statistics: Maximum
`ThreadpoolIndexRejected`	The number of rejected tasks in the index thread pool. If this number continually grows, consider scaling your cluster. Relevant statistics: Maximum
`ThreadpoolIndexThreads`	The size of the index thread pool. Relevant statistics: Maximum
`ThreadpoolSearchQueue`	The number of queued tasks in the search thread pool. If the queue size is consistently high, consider scaling your cluster. The maximum search queue size is 1,000. Relevant statistics: Maximum
`ThreadpoolSearchRejected`	The number of rejected tasks in the search thread pool. If this number continually grows, consider scaling your cluster. Relevant statistics: Maximum
`ThreadpoolSearchThreads`	The size of the search thread pool. Relevant statistics: Maximum
`ThreadpoolBulkQueue`	The number of queued tasks in the bulk thread pool. If the queue size is consistently high, consider scaling your cluster. Relevant statistics: Maximum
`ThreadpoolBulkRejected`	The number of rejected tasks in the bulk thread pool. If this number continually grows, consider scaling your cluster. Relevant statistics: Maximum
`ThreadpoolBulkThreads`	The size of the bulk thread pool. Relevant statistics: Maximum

Logging Amazon Elasticsearch Service Configuration API Calls with AWS CloudTrail

Amazon Elasticsearch Service integrates with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon ES. CloudTrail captures all configuration API calls for Amazon ES as events.

> **Note**
> CloudTrail only captures calls to the configuration API (p. 167), such as `CreateElasticsearchDomain` and `GetUpgradeStatus`, not the Elasticsearch APIs (p. 151), such as `_search` and `_bulk`.

The calls captured include calls from the Amazon ES console, CLI, or SDKs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Amazon ES. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Amazon ES, the IP address from which the request was made, who made the request, when it was made, and additional details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Amazon Elasticsearch Service Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Amazon ES, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Amazon ES, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All Amazon ES configuration API actions are logged by CloudTrail and are documented in the *Amazon ES Configuration API Reference* (p. 167).

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail userIdentity Element.

Understanding Amazon Elasticsearch Service Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateElasticsearchDomain` action.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/test-user",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "test-user",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-08-21T21:59:11Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2018-08-21T22:00:05Z",
  "eventSource": "es.amazonaws.com",
  "eventName": "CreateElasticsearchDomain",
  "awsRegion": "us-west-1",
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "elasticsearchVersion": "6.3",
    "elasticsearchClusterConfig": {
      "instanceType": "m4.large.elasticsearch",
      "instanceCount": 1
    },
    "snapshotOptions": {
      "automatedSnapshotStartHour": 0
    },
    "domainName": "test-domain",
    "encryptionAtRestOptions": {},
    "eBSOptions": {
      "eBSEnabled": true,
      "volumeSize": 10,
      "volumeType": "gp2"
    },
    "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}",
    "advancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    }
  },
  "responseElements": {
    "domainStatus": {
      "created": true,
      "elasticsearchClusterConfig": {
        "zoneAwarenessEnabled": false,
        "instanceType": "m4.large.elasticsearch",
        "dedicatedMasterEnabled": false,
        "instanceCount": 1
      },
      "cognitoOptions": {
        "enabled": false
      },
      "encryptionAtRestOptions": {
        "enabled": false
      },
      "advancedOptions": {
        "rest.action.multi.allow_explicit_index": "true"
      },
      "upgradeProcessing": false,
      "snapshotOptions": {
        "automatedSnapshotStartHour": 0
      },
      "eBSOptions": {
        "eBSEnabled": true,
        "volumeSize": 10,
        "volumeType": "gp2"
      },
      "elasticsearchVersion": "6.3",
      "processing": true,
      "aRN": "arn:aws:es:us-west-1:123456789012:domain/test-domain",
      "domainId": "123456789012/test-domain",
      "deleted": false,
      "domainName": "test-domain",
      "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":\"es:*\",\"Resource
\":\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}"
    }
  },
  "requestID": "12345678-1234-1234-1234-987654321098",
  "eventID": "87654321-4321-4321-4321-987654321098",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Tagging Amazon Elasticsearch Service Domains

You can use Amazon ES tags to add metadata to your Amazon ES domains. AWS does not apply any semantic meaning to your tags. Tags are interpreted strictly as character strings. All tags have the following elements.

Tag Element	Description
Tag key	The tag key is the required name of the tag. Tag keys must be unique for the Amazon ES domain to which they are attached. For a list of basic restrictions on tag keys and values, see User-Defined Tag Restrictions.
Tag value	The tag value is an optional string value of the tag. Tag values can be null and do not have to be unique in a tag set. For example, you can have a key-value pair in a tag set of project/Trinity and cost-center/Trinity. For a list of basic restrictions on tag keys and values, see User-Defined Tag Restrictions.

Each Amazon ES domain has a tag set, which contains all the tags that are assigned to that Amazon ES domain. AWS does not automatically set any tags on Amazon ES domains. A tag set can contain up to 50 tags, or it can be empty. If you add a tag to an Amazon ES domain that has the same key as an existing tag for a resource, the new value overwrites the old value.

You can use these tags to track costs by grouping expenses for similarly tagged resources. An Amazon ES domain tag is a name-value pair that you define and associate with an Amazon ES domain. The name is referred to as the *key*. You can use tags to assign arbitrary information to an Amazon ES domain. A tag key could be used, for example, to define a category, and the tag value could be an item in that category. For example, you could define a tag key of “project” and a tag value of “Salix,” indicating that the Amazon ES domain is assigned to the Salix project. You could also use tags to designate Amazon ES domains as being used for test or production by using a key such as environment=test or environment=production. We recommend that you use a consistent set of tag keys to make it easier to track metadata that is associated with Amazon ES domains.

You also can use tags to organize your AWS bill to reflect your own cost structure. To do this, sign up to get your AWS account bill with tag key values included. Then, organize your billing information according to resources with the same tag key values to see the cost of combined resources. For example, you can tag several Amazon ES domains with key-value pairs, and then organize your billing information to see the total cost for each domain across several services. For more information, see Using Cost Allocation Tags in the *AWS Billing and Cost Management* documentation.

> **Note**
> Tags are cached for authorization purposes. Because of this, additions and updates to tags on Amazon ES domains might take several minutes before they are available.

Working with Tags (Console)

Use the following procedure to create a resource tag.

To create a tag (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, choose your Amazon ES domain.
4. On the domain dashboard, choose **Manage tags**.
5. In the **Key** column, type a tag key.
6. (Optional) In the **Value** column, type a tag value.
7. Choose **Submit**.

To delete a tag (console)

Use the following procedure to delete a resource tag.

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, choose your Amazon ES domain.
4. On the domain dashboard, choose **Manage tags**.
5. Next to the tag that you want to delete, choose **Remove**.
6. Choose **Submit**.

For more information about using the console to work with tags, see Working with Tag Editor in the *AWS Management Console Getting Started Guide*.

Working with Tags (AWS CLI)

You can create resource tags using the AWS CLI with the **--add-tags** command.

Syntax

```
add-tags --arn=<domain_arn> --tag-list Key=<key>,Value=<value>
```

Parameter	Description
`--arn`	Amazon resource name for the Amazon ES domain to which the tag is attached.
`--tag-list`	Set of space-separated key-value pairs in the following format: `Key=<key>,Value=<value>`

Example

The following example creates two tags for the *logs* domain:

```
aws es add-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-list
 Key=service,Value=Elasticsearch Key=instances,Value=m3.2xlarge
```

You can remove tags from an Amazon ES domain using the **remove-tags** command.

Syntax

```
remove-tags --arn=<domain_arn> --tag-keys Key=<key>,Value=<value>
```

Parameter	Description
`--arn`	Amazon Resource Name (ARN) for the Amazon ES domain to which the tag is attached.
`--tag-keys`	Set of space-separated key-value pairs that you want to remove from the Amazon ES domain.

Example

The following example removes two tags from the *logs* domain that were created in the preceding example:

```
aws es remove-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-keys service
 instances
```

You can view the existing tags for an Amazon ES domain with the **list-tags** command:

Syntax

```
list-tags --arn=<domain_arn>
```

Parameter	Description
`--arn`	Amazon Resource Name (ARN) for the Amazon ES domain to which the tags are attached.

Example

The following example lists all resource tags for the *logs* domain:

```
aws es list-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs
```

Working with Tags (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon ES Configuration API Reference (p. 167), including the `AddTags`, `ListTags`, and `RemoveTags` operations. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Introduction to Indexing Data in Amazon Elasticsearch Service

Because Elasticsearch uses a REST API, numerous methods exist for indexing documents. You can use standard clients like curl or any programming language that can send HTTP requests. To further simplify the process of interacting with it, Elasticsearch has clients for many programming languages. Advanced users can skip directly to *Signing HTTP Requests* (p. 62).

For situations in which new data arrives incrementally (for example, customer orders from a small business), you might use the `_index` API to index documents as they arrive. For situations in which the flow of data is less frequent (for example, weekly updates to a marketing website), you might prefer to generate a file and send it to the `_bulk` API. For large numbers of documents, lumping requests together and using the `_bulk` API offers superior performance. If your documents are enormous, however, you might need to index them individually using the `_index` API.

For information about integrating data from other AWS services, see *Loading Streaming Data into Amazon ES* (p. 70).

Introduction to Indexing

Before you can search data, you must *index* it. Indexing is the method by which search engines organize data for fast retrieval. The resulting structure is called, fittingly, an index.

In Elasticsearch, the basic unit of data is a JSON *document*. Within an index, Elasticsearch organizes documents into *types* (arbitrary data categories that you define) and identifies them using a unique *ID*.

A request to the `_index` API looks like the following:

```
PUT elasticsearch_domain/index/type/id
{ "A JSON": "document" }
```

A request to the `_bulk` API looks a little different, because you specify the index, type, and ID in the bulk data:

```
POST elasticsearch_domain/_bulk
{ "index": { "_index" : "index", "_type" : "type", "_id" : "id" } }
{ "A JSON": "document" }
```

Bulk data must conform to a specific format, which requires a newline character (`\n`) at the end of every line, including the last line. This is the basic format:

```
action_and_metadata\n
optional_document\n
action_and_metadata\n
optional_document\n
...
```

For a short sample file, see the section called “Step 2: Uploading Data for Indexing” (p. 7).

Elasticsearch features automatic index creation when you add a document to an index that doesn't already exist. It also features automatic ID generation if you don't specify an ID in the request. This simple example automatically creates the `movies` index, establishes the document type of `movie`, indexes the document, and assigns it a unique ID:

```
POST elasticsearch_domain/movies/movie
{"title": "Spirited Away"}
```

Important

To use automatic ID generation, you must use the `POST` method instead of `PUT`.

To verify that the document exists, you can perform the following search:

```
GET elasticsearch_domain/movies/_search?pretty
```

The response should contain the following:

```
"hits" : {
  "total" : 1,
  "max_score" : 1.0,
  "hits" : [
    {
      "_index" : "movies",
      "_type" : "movie",
      "_id" : "AV4WaTnYxBoJaZkSFeX9",
      "_score" : 1.0,
      "_source" : {
        "title" : "Spirited Away"
      }
    }
  ]
}
```

Automatic ID generation has a clear downside: because the indexing code didn't specify a document ID, you can't easily update the document at a later time. To specify an ID of `7`, use the following request:

```
PUT elasticsearch_domain/movies/movie/7
{"title": "Spirited Away"}
```

Indices that you create in Elasticsearch versions 6.0 and later can only contain one document type. For best compatibility with future versions of Elasticsearch, use a single type, `_doc`, for all indices:

```
PUT elasticsearch_domain/more-movies/_doc/1
{"title": "Back to the Future"}
```

Indices default to five primary shards and one replica. If you want to specify non-default settings, create the index before adding documents:

```
PUT elasticsearch_domain/more-movies
{"settings": {"number_of_shards": 6, "number_of_replicas": 2}}
```

Note

For sample code, see *Signing HTTP Requests* (p. 62).

Elasticsearch indices have the following naming restrictions:

- All letters must be lowercase.

- Index names cannot begin with _ or -.
- Index names cannot contain spaces, commas, ", *, +, /, \, |, ?, #, >, or <.

Don't include sensitive information in index, type, or document ID names. Elasticsearch uses these names in its Uniform Resource Identifiers (URIs). Servers and applications often log HTTP requests, which can lead to unnecessary data exposure if URIs contain sensitive information:

```
2018-10-03T23:39:43 198.51.100.14 200 "GET https://elasticsearch_domain/dr-jane-doe/flu-
patients-2018/202-555-0100/ HTTP/1.1"
```

Even if you don't have permissions (p. 30) to view the associated JSON document, you could infer from this fake log line that one of Dr. Doe's patients with a phone number of 202-555-0100 had the flu in 2018.

Signing HTTP Requests to Amazon Elasticsearch Service

This chapter includes examples of how to send signed HTTP requests to Amazon Elasticsearch Service using Elasticsearch clients and other common libraries. These code samples are for interacting with the Elasticsearch APIs, such as `_index`, `_bulk`, and `_snapshot`.

> **Important**
> For examples of how to interact with the Configuration API, including operations like creating, updating, and deleting Amazon ES domains, see *Using the AWS SDKs* (p. 163).

Topics

Java

The easiest way of sending a signed request is to use the AWS Request Signing Interceptor. The repository contains some samples to help you get started. The following example uses the Elasticsearch low-level Java REST client to perform two unrelated actions: registering a snapshot repository and indexing a document. You must provide values for `region` and `host`.

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpHost;
import org.apache.http.HttpRequestInterceptor;
import org.apache.http.entity.ContentType;
import org.apache.http.nio.entity.NStringEntity;
import org.elasticsearch.client.Response;
import org.elasticsearch.client.RestClient;
import com.amazonaws.auth.AWS4Signer;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.http.AWSRequestSigningApacheInterceptor;
import java.io.IOException;
import java.util.Collections;
import java.util.Map;

public class AmazonElasticsearchServiceSample {

    private static String serviceName = "es";
    private static String region = "us-west-1";
    private static String aesEndpoint = "https://domain.us-west-1.es.amazonaws.com";

    private static String payload = "{ \"type\": \"s3\", \"settings\":
 { \"bucket\": \"your-bucket\", \"region\": \"us-west-1\", \"role_arn\":
 \"arn:aws:iam::123456789012:role/TheServiceRole\" } }";
    private static String snapshotPath = "/_snapshot/my-snapshot-repo";

    private static String sampleDocument = "{" + "\"title\":\"Walk the Line\"," +
 "\"director\":\"James Mangold\"," + "\"year\":\"2005\"}";
```

```
    private static String indexingPath = "/my-index/_doc";

    static final AWSCredentialsProvider credentialsProvider = new
 DefaultAWSCredentialsProviderChain();

    public static void main(String[] args) throws IOException {
        RestClient esClient = esClient(serviceName, region);

        // Register a snapshot repository
        HttpEntity entity = new NStringEntity(payload, ContentType.APPLICATION_JSON);
        Map<String, String> params = Collections.emptyMap();
        Response response = esClient.performRequest("PUT", snapshotPath, params, entity);
        System.out.println(response.toString());

        // Index a document
        entity = new NStringEntity(sampleDocument, ContentType.APPLICATION_JSON);
        String id = "1";
        response = esClient.performRequest("PUT", indexingPath + "/" + id, params, entity);
        System.out.println(response.toString());
    }

    // Adds the interceptor to the ES REST client
    public static RestClient esClient(String serviceName, String region) {
        AWS4Signer signer = new AWS4Signer();
        signer.setServiceName(serviceName);
        signer.setRegionName(region);
        HttpRequestInterceptor interceptor = new
 AWSRequestSigningApacheInterceptor(serviceName, signer, credentialsProvider);
        return
 RestClient.builder(HttpHost.create(aesEndpoint)).setHttpClientConfigCallback(hacb ->
 hacb.addInterceptorLast(interceptor)).build();
    }
}
```

If you prefer the high-level REST client, which offers most of the same features and simpler code, try the following sample, which also uses the AWS Request Signing Interceptor:

```
import org.apache.http.HttpHost;
import org.apache.http.HttpRequestInterceptor;
import org.elasticsearch.action.index.IndexRequest;
import org.elasticsearch.action.index.IndexResponse;
import org.elasticsearch.client.RestClient;
import org.elasticsearch.client.RestHighLevelClient;
import com.amazonaws.auth.AWS4Signer;
import com.amazonaws.auth.AWSCredentialsProvider;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.http.AWSRequestSigningApacheInterceptor;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

public class AmazonElasticsearchServiceSample {

    private static String serviceName = "es";
    private static String region = "us-west-1";
    private static String aesEndpoint = ""; // e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
    private static String index = "my-index";
    private static String type = "_doc";
    private static String id = "1";

    static final AWSCredentialsProvider credentialsProvider = new
 DefaultAWSCredentialsProviderChain();
```

```
    public static void main(String[] args) throws IOException {
        RestHighLevelClient esClient = esClient(serviceName, region);

        // Create the document as a hash map
        Map<String, Object> document = new HashMap<>();
        document.put("title", "Walk the Line");
        document.put("director", "James Mangold");
        document.put("year", "2005");

        // Form the indexing request, send it, and print the response
        IndexRequest request = new IndexRequest(index, type, id).source(document);
        IndexResponse response = esClient.index(request);
        System.out.println(response.toString());
    }

    // Adds the interceptor to the ES REST client
    public static RestHighLevelClient esClient(String serviceName, String region) {
        AWS4Signer signer = new AWS4Signer();
        signer.setServiceName(serviceName);
        signer.setRegionName(region);
        HttpRequestInterceptor interceptor = new
 AWSRequestSigningApacheInterceptor(serviceName, signer, credentialsProvider);
        return new
 RestHighLevelClient(RestClient.builder(HttpHost.create(aesEndpoint)).setHttpClientConfigCallback(hacb
 -> hacb.addInterceptorLast(interceptor)));
    }
}
```

Tip
Both signed samples use the default credential chain. Run `aws configure` using the AWS CLI to set your credentials.

Python

You can install elasticsearch-py, the Elasticsearch client for Python, using pip. Instead of the client, you might prefer requests. The requests-aws4auth and SDK for Python (Boto 3) packages simplify the authentication process, but are not strictly required. From the terminal, run the following commands:

```
pip install boto3
pip install elasticsearch
pip install requests
pip install requests-aws4auth
```

The following sample code establishes a secure connection to the specified Amazon ES domain and indexes a single document using the `_index` API. You must provide values for `region` and `host`.

```
from elasticsearch import Elasticsearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3

host = '' # For example, my-test-domain.us-east-1.es.amazonaws.com
region = '' # e.g. us-west-1

service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service)

es = Elasticsearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    connection_class = RequestsHttpConnection
)

document = {
    "title": "Moneyball",
    "director": "Bennett Miller",
    "year": "2011"
}

es.index(index="movies", doc_type="_doc", id="5", body=document)

print(es.get(index="movies", doc_type="_doc", id="5"))
```

If you don't want to use elasticsearch-py, you can just make standard HTTP requests. This sample creates a new index with seven shards and two replicas:

```
from requests_aws4auth import AWS4Auth
import boto3
import requests

host = '' # The domain with https:// and trailing slash. For example, https://my-test-
domain.us-east-1.es.amazonaws.com/
path = 'my-index' # the Elasticsearch API endpoint
region = '' # For example, us-west-1

service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service)

url = host + path

# The JSON body to accompany the request (if necessary)
payload = {
    "settings" : {
        "number_of_shards" : 7,
        "number_of_replicas" : 2
    }
}

r = requests.put(url, auth=awsauth, json=payload) # requests.get, post, and delete have
 similar syntax

print(r.text)
```

This next example uses the Beautiful Soup library to help build a bulk file from a local directory of HTML files. Using the same client as the first example, you can send the file to the `_bulk` API for indexing. You could use this code as the basis for adding search functionality to a website:

```
from bs4 import BeautifulSoup
from elasticsearch import Elasticsearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3
import glob
import json

bulk_file = ''
id = 1

# This loop iterates through all HTML files in the current directory and
# indexes two things: the contents of the first h1 tag and all other text.
```

```
for html_file in glob.glob('*.htm'):

    with open(html_file) as f:
        soup = BeautifulSoup(f, 'html.parser')

    title = soup.h1.string
    body = soup.get_text(" ", strip=True)
    # If get_text() is too noisy, you can do further processing on the string.

    index = { 'title': title, 'body': body, 'link': html_file }
    # If running this script on a website, you probably need to prepend the URL and path to
 html_file.

    # The action_and_metadata portion of the bulk file
    bulk_file += '{ "index" : { "_index" : "site", "_type" : "_doc", "_id" : "' + str(id) +
 '" } }\n'

    # The optional_document portion of the bulk file
    bulk_file += json.dumps(index) + '\n'

    id += 1

host = '' # For example, my-test-domain.us-east-1.es.amazonaws.com
region = '' # e.g. us-west-1

service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service)

es = Elasticsearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    connection_class = RequestsHttpConnection
)

es.bulk(bulk_file)

print(es.search(q='some test query'))
```

Ruby

This first example uses the Elasticsearch Ruby client and Faraday middleware to perform the request signing. From the terminal, run the following commands:

```
gem install elasticsearch
gem install faraday_middleware-aws-sigv4
```

This sample code creates a new Elasticsearch client, configures Faraday middleware to sign requests, and indexes a single document. You must provide values for `host` and `region`.

```
require 'elasticsearch'
require 'faraday_middleware/aws_sigv4'

host = '' # e.g. https://my-domain.region.es.com
index = 'ruby-index'
type = '_doc'
id = '1'
document = {
  year: 2007,
  title: '5 Centimeters per Second',
  info: {
    plot: 'Told in three interconnected segments, we follow a young man named Takaki
 through his life.',
    rating: 7.7
  }
}

region = '' # e.g. us-west-1
service = 'es'

client = Elasticsearch::Client.new(url: host) do |f|
  f.request :aws_sigv4,
    service: service,
    region: region,
    access_key_id: ENV['AWS_ACCESS_KEY_ID'],
    secret_access_key: ENV['AWS_SECRET_ACCESS_KEY'],
    session_token: ENV['AWS_SESSION_TOKEN'] # optional
end

puts client.index index: index, type: type, id: id, body: document
```

If your credentials don't work, export them at the terminal using the following commands:

```
export AWS_ACCESS_KEY_ID="your-access-key"
export AWS_SECRET_ACCESS_KEY="your-secret-key"
export AWS_SESSION_TOKEN=""your-session-token"
```

This next example uses the AWS SDK for Ruby and standard Ruby libraries to send a signed HTTP request. Like the first example, it indexes a single document. You must provide values for host and region.

```
require 'aws-sdk-elasticsearchservice'

host = '' # e.g. https://my-domain.region.es.com
index = 'ruby-index'
type = '_doc'
id = '2'
document = {
  year: 2007,
  title: '5 Centimeters per Second',
  info: {
    plot: 'Told in three interconnected segments, we follow a young man named Takaki
 through his life.',
    rating: 7.7
  }
}

service = 'es'
region = '' # e.g. us-west-1

signer = Aws::Sigv4::Signer.new(
  service: service,
  region: region,
  access_key_id: ENV['AWS_ACCESS_KEY_ID'],
  secret_access_key: ENV['AWS_SECRET_ACCESS_KEY'],
  session_token: ENV['AWS_SESSION_TOKEN']
)

signature = signer.sign_request(
  http_method: 'PUT',
  url: host + '/' + index + '/' + type + '/' + id,
  body: document.to_json
)

uri = URI(host + '/' + index + '/' + type + '/' + id)

Net::HTTP.start(uri.host, uri.port, :use_ssl => true) do |http|
  request = Net::HTTP::Put.new uri
  request.body = document.to_json
  request['Host'] = signature.headers['host']
  request['X-Amz-Date'] = signature.headers['x-amz-date']
  request['X-Amz-Security-Token'] = signature.headers['x-amz-security-token']
  request['X-Amz-Content-Sha256']= signature.headers['x-amz-content-sha256']
  request['Authorization'] = signature.headers['authorization']
  request['Content-Type'] = 'application/json'
  response = http.request request
  puts response.body
end
```

Node

This example uses the SDK for JavaScript in Node.js. From the terminal, run the following commands:

```
npm install aws-sdk
```

This sample code indexes a single document. You must provide values for `region` and `domain`.

```
var AWS = require('aws-sdk');

var region = ''; // e.g. us-west-1
var domain = ''; // e.g. search-domain.region.es.amazonaws.com
var index = 'node-test';
var type = '_doc';
var id = '1';
var json = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

indexDocument(json);

function indexDocument(document) {
  var endpoint = new AWS.Endpoint(domain);
  var request = new AWS.HttpRequest(endpoint, region);

  request.method = 'PUT';
  request.path += index + '/' + type + '/' + id;
  request.body = JSON.stringify(document);
  request.headers['host'] = domain;
  request.headers['Content-Type'] = 'application/json';

  var credentials = new AWS.EnvironmentCredentials('AWS');
  var signer = new AWS.Signers.V4(request, 'es');
  signer.addAuthorization(credentials, new Date());

  var client = new AWS.HttpClient();
  client.handleRequest(request, null, function(response) {
    console.log(response.statusCode + ' ' + response.statusMessage);
    var responseBody = '';
    response.on('data', function (chunk) {
      responseBody += chunk;
    });
    response.on('end', function (chunk) {
      console.log('Response body: ' + responseBody);
    });
  }, function(error) {
    console.log('Error: ' + error);
  });
}
```

If your credentials don't work, export them at the terminal using the following commands:

```
export AWS_ACCESS_KEY_ID="your-access-key"
export AWS_SECRET_ACCESS_KEY="your-secret-key"
export AWS_SESSION_TOKEN=""your-session-token"
```

Loading Streaming Data into Amazon Elasticsearch Service

You can load streaming data into your Amazon Elasticsearch Service domain from many different sources. Some sources, like Amazon Kinesis Data Firehose and Amazon CloudWatch Logs, have built-in support for Amazon ES. Others, like Amazon S3, Amazon Kinesis Data Streams, and Amazon DynamoDB, use AWS Lambda functions as event handlers. The Lambda functions respond to new data by processing it and streaming it to your domain.

> **Note**
> Lambda supports several popular programming languages and is available in most AWS Regions. For more information, see Building Lambda Functions in the *AWS Lambda Developer Guide* and AWS Lambda Regions in the *AWS General Reference*.

Topics

Loading Streaming Data into Amazon ES from Amazon S3

You can use Lambda to send data to your Amazon ES domain from Amazon S3. New data that arrives in an S3 bucket triggers an event notification to Lambda, which then runs your custom code to perform the indexing.

This method of streaming data is extremely flexible. You can index object metadata, or if the object is plaintext, parse and index some elements of the object body. This section includes some unsophisticated Python sample code that uses regular expressions to parse a log file and index the matches.

> **Tip**
> For more robust code in Node.js, see amazon-elasticsearch-lambda-samples on GitHub. Some Lambda blueprints also contain useful parsing examples.

Prerequisites

Before proceeding, you must have the following resources.

Prerequisite	Description
Amazon S3 Bucket	For more information, see Creating a Bucket in the *Amazon Simple Storage Service Getting Started Guide*. The bucket must reside in the same region as your Amazon ES domain.
Amazon ES Domain	The destination for data after your Lambda function processes it. For more information, see Creating Amazon ES Domains (p. 10).

Creating the Lambda Deployment Package

Deployment packages are ZIP or JAR files that contain your code and its dependencies. This section includes Python sample code. For other programming languages, see Creating a Deployment Package in the *AWS Lambda Developer Guide*.

1. Create a directory. In this sample, we use the name `s3-to-es`.
2. Create a file in the directory named `sample.py`:

```
import boto3
import re
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the Amazon ES domain, including https://
index = 'lambda-s3-index'
type = 'lambda-type'
url = host + '/' + index + '/' + type

headers = { "Content-Type": "application/json" }

s3 = boto3.client('s3')

# Regular expressions used to parse some simple log lines
ip_pattern = re.compile('(\d+\.\d+\.\d+\.\d+)')
time_pattern = re.compile('\[(\d+\/\w\w\w\/\d\d\d\d:\d\d:\d\d:\d\d\s-\d\d\d\d)\]')
message_pattern = re.compile('\"(.+)\"')

# Lambda execution starts here
def handler(event, context):
    for record in event['Records']:

        # Get the bucket name and key for the new file
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']

        # Get, read, and split the file into lines
        obj = s3.get_object(Bucket=bucket, Key=key)
        body = obj['Body'].read()
        lines = body.splitlines()

        # Match the regular expressions to each line and index the JSON
        for line in lines:
            ip = ip_pattern.search(line).group(1)
            timestamp = time_pattern.search(line).group(1)
            message = message_pattern.search(line).group(1)

            document = { "ip": ip, "timestamp": timestamp, "message": message }
            r = requests.post(url, auth=awsauth, json=document, headers=headers)
```

   Edit the variables for `region` and `host`.

3. Install dependencies:

```
cd s3-to-es
pip install requests -t .
pip install requests_aws4auth -t .
```

All Lambda execution environments have Boto3 installed, so you don't need to include it in your deployment package.

Tip
If you use macOS, these commands might not work properly. As a workaround, add a file named `setup.cfg` to the `s3-to-es` directory:

```
[install]
prefix=
```

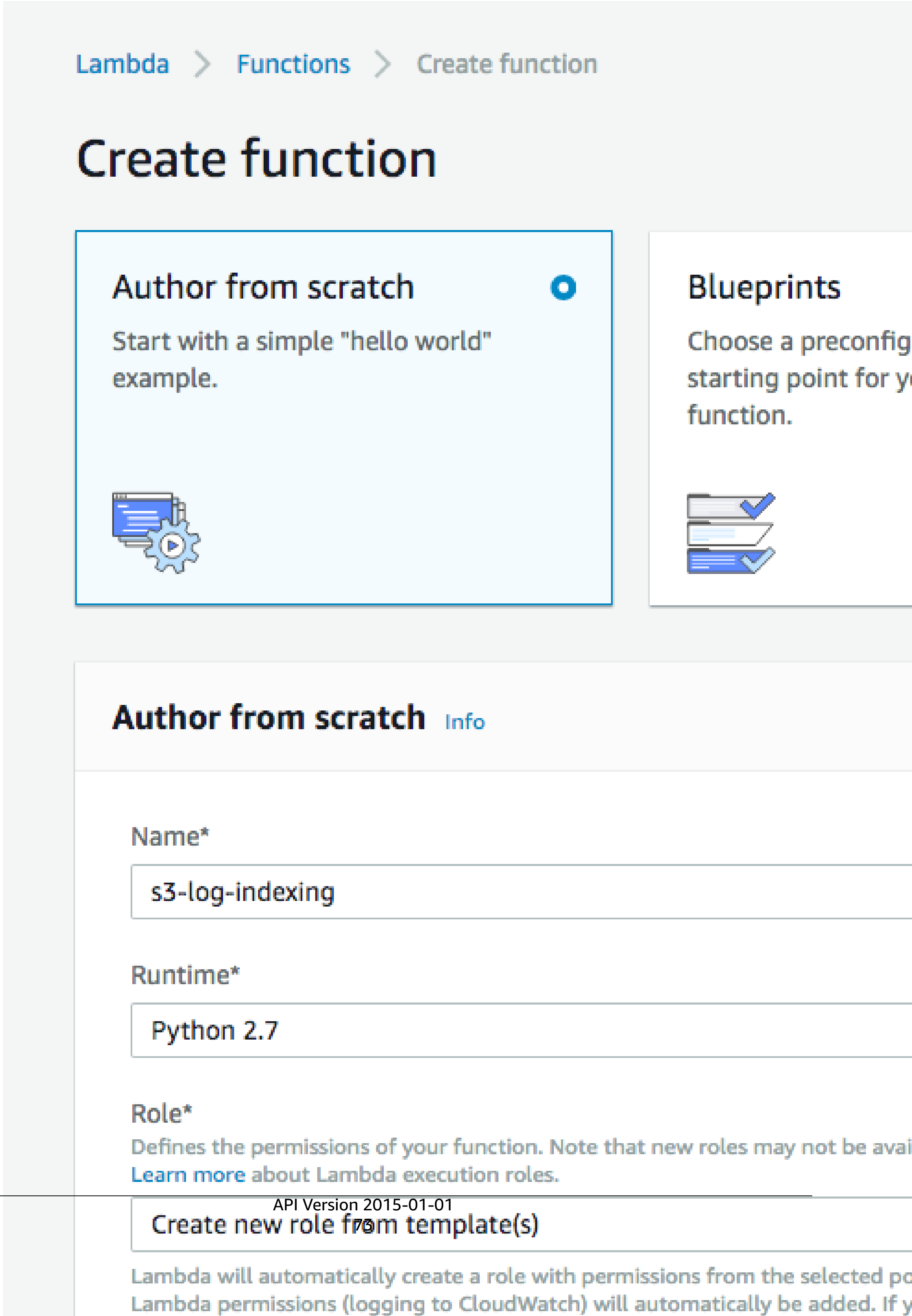
4. Package the application code and dependencies:

```
zip -r lambda.zip *
```

Creating the Lambda Function

After you create the deployment package, you can create the Lambda function. When you create a function, choose a name, runtime (for example, Python 2.7), and IAM role. The IAM role defines the permissions for your function. For detailed instructions, see Create a Simple Lambda Function in the *AWS Lambda Developer Guide*.

This example assumes that you are using the console. Choose Python 2.7 and a role that has S3 read permissions and Amazon ES write permissions, as shown in the following screenshot.



Create function

Author from scratch ●

Start with a simple "hello world" example.

Blueprints

Choose a preconfig
starting point for y
function.

Author from scratch

Name*

s3-log-indexing

Runtime*

Python 2.7

Role*

Defines the permissions of your function. Note that new roles may not be avai
Learn more about Lambda execution roles.

Create new role from template(s)

Lambda will automatically create a role with permissions from the selected po
Lambda permissions (logging to CloudWatch) will automatically be added. If y

After you create the function, you must add a trigger. For this example, we want the code to execute whenever a log file arrives in an S3 bucket:

1. Choose S3.
2. Choose your bucket.
3. For **Event type**, choose **PUT**.
4. For **Prefix**, type `logs/`.
5. For **Filter pattern**, type `.log`.
6. Select **Enable trigger**.
7. Choose **Add**.

Finally, you can upload your deployment package:

1. For **Handler**, type `sample.handler`. This setting tells Lambda the file (`sample.py`) and method (`handler`) that it should execute after a trigger.
2. For **Code entry type**, choose **Upload a .ZIP file**, and then follow the prompts to upload your deployment package.
3. Choose **Save**.

At this point, you have a complete set of resources: a bucket for log files, a function that executes whenever a log file is added to the bucket, code that performs the parsing and indexing, and an Amazon ES domain for searching and visualization.

Testing the Lambda Function

After you create the function, you can test it by uploading a file to the Amazon S3 bucket. Create a file named `sample.log` using following sample log lines:

```
12.345.678.90 - [10/Oct/2000:13:55:36 -0700] "PUT /some-file.jpg"
12.345.678.91 - [10/Oct/2000:14:56:14 -0700] "GET /some-file.jpg"
```

Upload the file to the `logs` folder of your S3 bucket. For instructions, see Add an Object to a Bucket in the *Amazon Simple Storage Service Getting Started Guide*.

Then use the Amazon ES console or Kibana to verify that the `lambda-s3-index` index contains two documents. You can also make a standard search request:

```
GET https://es-domain/lambda-index/_search?pretty
{
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "lambda-s3-index",
        "_type" : "lambda-type",
        "_id" : "vTYXaWIBJWV_TTkEuSDg",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.91",
          "message" : "GET /some-file.jpg",
          "timestamp" : "10/Oct/2000:14:56:14 -0700"
        }
      },
      {
        "_index" : "lambda-s3-index",
        "_type" : "lambda-type",
        "_id" : "vjYmaWIBJWV_TTkEuCAB",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.90",
          "message" : "PUT /some-file.jpg",
          "timestamp" : "10/Oct/2000:13:55:36 -0700"
        }
      }
    ]
  }
}
```

Loading Streaming Data into Amazon ES from Amazon Kinesis Data Streams

You can load streaming data from Kinesis Data Streams to Amazon ES. New data that arrives in the data stream triggers an event notification to Lambda, which then runs your custom code to perform the indexing. This section includes some unsophisticated Python sample code. For more robust code in Node.js, see amazon-elasticsearch-lambda-samples on GitHub.

Prerequisites

Before proceeding, you must have the following resources.

<table>
<tr><th>Prerequisite</th><th>Description</th></tr>
<tr><td>Amazon Kinesis Data Stream</td><td>The event source for your Lambda function. To learn more, see Kinesis Data Streams.</td></tr>
<tr><td>Amazon ES Domain</td><td>The destination for data after your Lambda function processes it. For more information, see Creating Amazon ES Domains (p. 10).</td></tr>
<tr><td>IAM Role</td><td>This role must have basic Amazon ES, Kinesis, and Lambda permissions, such as the following:
<pre>
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpPost",
        "es:ESHttpPut",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:DescribeStream",
        "kinesis:ListStreams"
      ],
      "Resource": "*"
    }
  ]
}
</pre></td></tr>
</table>

<table>
<tr><th>Prerequisite</th><th>Description</th></tr>
<tr><td></td><td>

The role must have the following trust relationship:

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

To learn more, see Creating IAM Roles in the *IAM User Guide*.

</td></tr>
</table>

Creating the Lambda Function

Follow the instructions in the section called “Creating the Lambda Deployment Package” (p. 71), but create a directory named `kinesis-to-es` and use the following code for `sample.py`:

```
import base64
import boto3
import json
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the Amazon ES domain, including https://
index = 'lambda-kine-index'
type = 'lambda-kine-type'
url = host + '/' + index + '/' + type + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        id = record['eventID']
        timestamp = record['kinesis']['approximateArrivalTimestamp']

        # Kinesis data is base64-encoded, so decode here
        message = base64.b64decode(record['kinesis']['data'])

        # Create the JSON document
        document = { "id": id, "timestamp": timestamp, "message": message }
        # Index the document
        r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return 'Processed ' + str(count) + ' items.'
```

Edit the variables for `region` and `host`.

Use the following commands to install your dependencies:

```
cd kinesis-to-es
pip install requests -t .
pip install requests_aws4auth -t .
```

Then follow the instructions in the section called “Creating the Lambda Function” (p. 72), but specify the IAM role from the section called “Prerequisites” (p. 75) and the following settings for the trigger:

- **Kinesis stream**: your Kinesis stream
- **Batch size**: 100
- **Starting position**: Trim horizon

To learn more, see Working with Amazon Kinesis Data Streams in the *Amazon Kinesis Data Streams Developer Guide*.

At this point, you have a complete set of resources: a Kinesis data stream, a function that executes after the stream receives new data and indexes that data, and an Amazon ES domain for searching and visualization.

Testing the Lambda Function

After you create the function, you can test it by adding a new record to the data stream using the AWS CLI:

```
aws kinesis put-record --stream-name es-test --data "My test data." --partition-key
 partitionKey1 --region us-west-1
```

Then use the Amazon ES console or Kibana to verify that `lambda-kine-index` contains a document. You can also use the following request:

```
GET https://es-domain/lambda-kine-index/_search
{
  "hits" : [
    {
      "_index": "lambda-kine-index",
      "_type": "lambda-kine-type",
      "_id":
 "shardId-000000000000:49583511615762699495012960821421456686529436680496087042",
      "_score": 1,
      "_source": {
        "timestamp": 1523648740.051,
        "message": "My test data.",
        "id":
 "shardId-000000000000:49583511615762699495012960821421456686529436680496087042"
      }
    }
  ]
}
```

Loading Streaming Data into Amazon ES from Amazon DynamoDB

You can use AWS Lambda to send data to your Amazon ES domain from Amazon DynamoDB. New data that arrives in the database table triggers an event notification to Lambda, which then runs your custom code to perform the indexing.

Prerequisites

Before proceeding, you must have the following resources.

<table>
<tr><th>Prerequisite</th><th>Description</th></tr>
<tr><td>DynamoDB Table</td><td>The table contains your source data. For more information, see Basic Operations for Tables in the <em>Amazon DynamoDB Developer Guide</em>.
The table must reside in the same region as your Amazon ES domain and have a stream set to <strong>New image</strong>. To learn more, see Enabling a Stream.</td></tr>
<tr><td>Amazon ES Domain</td><td>The destination for data after your Lambda function processes it. For more information, see Creating Amazon ES Domains (p. 10).</td></tr>
<tr><td>IAM Role</td><td>This role must have basic Amazon ES, DynamoDB, and Lambda execution permissions, such as the following:
<pre>
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpPost",
        "es:ESHttpPut",
        "dynamodb:DescribeStream",
        "dynamodb:GetRecords",
        "dynamodb:GetShardIterator",
        "dynamodb:ListStreams",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
</pre>
The role must have the following trust relationship:
<pre>
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
</pre></td></tr>
</table>

Prerequisite	Description

	`    }`
	`  ]`
	`}`
	To learn more, see Creating IAM Roles in the *IAM User Guide*.

Creating the Lambda Function

Follow the instructions in the section called "Creating the Lambda Deployment Package" (p. 71), but create a directory named `ddb-to-es` and use the following code for `sample.py`:

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-east-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the Amazon ES domain, with https://
index = 'lambda-index'
type = 'lambda-type'
url = host + '/' + index + '/' + type + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        # Get the primary key for use as the Elasticsearch ID
        id = record['dynamodb']['Keys']['id']['S']

        if record['eventName'] == 'REMOVE':
            r = requests.delete(url + id, auth=awsauth)
        else:
            document = record['dynamodb']['NewImage']
            r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return str(count) + ' records processed.'
```

Edit the variables for `region` and `host`.

Use the following commands to install your dependencies:

```
cd ddb-to-es
pip install requests -t .
pip install requests_aws4auth -t .
```

Then follow the instructions in the section called "Creating the Lambda Function" (p. 72), but specify the IAM role from the section called "Prerequisites" (p. 78) and the following settings for the trigger:

- **Table**: your DynamoDB table
- **Batch size**: 100
- **Starting position**: Trim horizon

To learn more, see Processing New Items in a DynamoDB Table in the *Amazon DynamoDB Developer Guide*.

At this point, you have a complete set of resources: a DynamoDB table for your source data, a DynamoDB stream of changes to the table, a function that executes after your source data changes and indexes those changes, and an Amazon ES domain for searching and visualization.

Testing the Lambda Function

After you create the function, you can test it by adding a new item to the DynamoDB table using the AWS CLI:

```
aws dynamodb put-item --table-name es-test --item '{"director": {"S": "Kevin
 Costner"},"id": {"S": "00001"},"title": {"S": "The Postman"}}' --region us-west-1
```

Then use the Amazon ES console or Kibana to verify that `lambda-index` contains a document. You can also use the following request:

```
GET https://es-domain/lambda-index/lambda-type/00001
{
    "_index": "lambda-index",
    "_type": "lambda-type",
    "_id": "00001",
    "_version": 1,
    "found": true,
    "_source": {
        "director": {
            "S": "Kevin Costner"
        },
        "id": {
            "S": "00001"
        },
        "title": {
            "S": "The Postman"
        }
    }
}
```

Loading Streaming Data into Amazon ES from Amazon Kinesis Data Firehose

Kinesis Data Firehose supports Amazon ES as a delivery destination. For instructions about how to load streaming data into Amazon ES, see Creating a Kinesis Data Firehose Delivery Stream and Choose Amazon ES for Your Destination in the *Amazon Kinesis Data Firehose Developer Guide*.

Before you load data into Amazon ES, you might need to perform transforms on the data. To learn more about using Lambda functions to perform this task, see Data Transformation in the same guide.

As you configure a delivery stream, Kinesis Data Firehose features a "one-click" IAM role that gives it the resource access it needs to send data to Amazon ES, back up data on Amazon S3, and transform data using Lambda. Because of the complexity involved in creating such a role manually, we recommend using the provided role.

Loading Streaming Data into Amazon ES from Amazon CloudWatch

You can load streaming data from CloudWatch Logs to your Amazon ES domain by using a CloudWatch Logs subscription. For information about Amazon CloudWatch subscriptions, see Real-time Processing of Log Data with Subscriptions. For configuration information, see Streaming CloudWatch Logs Data to Amazon Elasticsearch Service in the *Amazon CloudWatch Developer Guide*.

Loading Data into Amazon ES from AWS IoT

You can send data from AWS IoT using rules. To learn more, see Amazon ES Action in the *AWS IoT Developer Guide*.

Searching Data in Amazon Elasticsearch Service

As you might expect from a search engine, Elasticsearch offers numerous options for searching your data. This chapter introduces a few common ways of performing searches with Amazon ES. You can use Postman to test the various requests. For code samples that send signed HTTP requests to Amazon ES, see *Signing HTTP Requests* (p. 62).

> **Note**
> All example requests in this chapter work with the Elasticsearch 6.*x* APIs. Some requests might not work with older Elasticsearch versions.

URI Searches

Universal Resource Identifier (URI) searches are the simplest form of search. In a URI search, you specify the query as an HTTP request parameter:

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/_search?q=house
```

A sample response might look like the following:

```
{
  "took": 25,
  "timed_out": false,
  "_shards": {
    "total": 10,
    "successful": 10,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 85,
    "max_score": 6.6137657,
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "tt0077975",
        "_score": 6.6137657,
        "_source": {
          "directors": [
            "John Landis"
          ],
          "release_date": "1978-07-27T00:00:00Z",
          "rating": 7.5,
          "genres": [
            "Comedy",
            "Romance"
          ],
          "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTY2OTQxNTc1OF5BMl5BanBnXkFtZTYwNjA3NjI5._V1_SX400_.jpg",
          "plot": "At a 1962 College, Dean Vernon Wormer is determined to expel the entire
 Delta Tau Chi Fraternity, but those troublemakers have other plans for him.",
          "title": "Animal House",
          "rank": 527,
          "running_time_secs": 6540,
          "actors": [
            "John Belushi",
            "Karen Allen",
            "Tom Hulce"
          ],
          "year": 1978,
          "id": "tt0077975"
        }
      },
      ...
    ]
  }
}
```

By default, this query searches all fields of all indices for the term *house*. To narrow the search, specify an index (`movies`) and a document field (`title`) in the URI:

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?q=title:house
```

You can include additional parameters in the request, but the supported parameters provide only a small subset of the Elasticsearch search options. The following request returns 20 results (instead of the default of 10) and sorts by year (rather than by `_score`):

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?
q=title:house&size=20&sort=year:desc
```

Request Body Searches

To perform more complex searches, use the HTTP request body and the Elasticsearch domain-specific language (DSL) for queries. The query DSL lets you specify the full range of Elasticsearch search options. The following `match` query is similar to the final URI search (p. 82) example:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "sort": {
    "year": {
      "order": "desc"
    }
  },
  "query": {
    "query_string": {
      "default_field": "title",
      "query": "house"
    }
  }
}
```

> **Note**
> The `_search` API accepts HTTP `GET` and `POST` for request body searches, but not all HTTP clients support adding a request body to a `GET` request. `POST` is the more universal choice.

In many cases, you might want to search several fields, but not all fields. Use the `multi_match` query:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
```

```
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title", "plot", "actors", "directors"]
    }
  }
}
```

Boosting Fields

You can improve search relevancy by "boosting" certain fields. Boosts are multipliers that weigh matches in one field more heavily than matches in other fields. In the following example, a match for *john* in the `title` field influences `_score` twice as much as a match in the `plot` field and four times as much as a match in the `actors` or `directors` fields. The result is that films like *John Wick* and *John Carter* are near the top of the search results, and films starring John Travolta are near the bottom.

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "john",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

Paginating Search Results

If you need to display a large number of search results, you can implement pagination using the `from` parameter. The following request returns results 20–39 of the zero-indexed list of search results:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "from": 20,
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

Search Result Highlighting

The `highlight` option tells Elasticsearch to return an additional object inside of the `hits` array if the query matched one or more fields:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    }
  }
}
```

If the query matched the content of the `plot` field, a hit might look like the following:

```
{
  "_index": "movies",
  "_type": "movie",
  "_id": "tt0091541",
  "_score": 11.276199,
  "_source": {
    "directors": [
      "Richard Benjamin"
    ],
    "release_date": "1986-03-26T00:00:00Z",
    "rating": 6,
    "genres": [
      "Comedy",
      "Music"
    ],
    "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTIzODEzODE2OF5BMl5BanBnXkFtZTcwNjQ3ODcyMQ@@._V1_SX400_.jpg",
    "plot": "A young couple struggles to repair a hopelessly dilapidated house.",
    "title": "The Money Pit",
    "rank": 4095,
    "running_time_secs": 5460,
    "actors": [
      "Tom Hanks",
      "Shelley Long",
      "Alexander Godunov"
    ],
    "year": 1986,
    "id": "tt0091541"
  },
  "highlight": {
    "plot": [
      "A young couple struggles to repair a hopelessly dilapidated <em>house</em>."
    ]
  }
}
```

By default, Elasticsearch wraps the matching string in `<em>` tags, provides up to 100 characters of context around the match, and breaks content into sentences by identifying punctuation marks, spaces, tabs, and line breaks. All of these settings are customizable:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    },
    "pre_tags": "<strong>",
    "post_tags": "</strong>",
    "fragment_size": 200,
    "boundary_chars": ".,!? "
  }
}
```

Count API

If you're not interested in the contents of your documents and just want to know the number of matches, you can use the `_count` API instead of the `_search` API. The following request uses the `query_string` query to identify romantic comedies:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_count
{
  "query": {
    "query_string": {
      "default_field": "genres",
      "query": "romance AND comedy"
    }
  }
}
```

A sample response might look like the following:

```
{
  "count": 564,
  "_shards": {
    "total": 5,
    "successful": 5,
    "skipped": 0,
    "failed": 0
  }
}
```

Working with Amazon Elasticsearch Service Index Snapshots

Snapshots are backups of a cluster's data and state. State includes cluster settings, node information, index settings, and shard allocation.

Snapshots provide a convenient way to migrate data across Amazon Elasticsearch Service domains and recover from failure. The service supports restoring from snapshots taken on both Amazon ES domains and self-managed Elasticsearch clusters.

Amazon ES takes daily automated snapshots of the primary index shards in a domain, as described in the section called "Configuring Automatic Snapshots" (p. 23). The service stores up to 14 of these snapshots for no more than 30 days in a preconfigured Amazon S3 bucket at no additional charge to you. You can use these snapshots to restore the domain.

If the cluster enters red status and you don't correct the problem, you start to lose automated snapshots after 16 days. For troubleshooting steps, see the section called "Red Cluster Status" (p. 143).

You cannot use automated snapshots to migrate to new domains. Automated snapshots are read-only from within a given domain. For migrations, you must use manual snapshots stored in your own repository (an S3 bucket). Standard S3 charges apply to manual snapshots.

Tip

Many users find tools like Curator convenient for index and snapshot management. Use pip to install Curator:

```
pip install elasticsearch-curator
```

Curator offers advanced filtering functionality that can help simplify management tasks on complex clusters. Amazon ES supports Curator on domains running Elasticsearch version 5.1 and above. You can use Curator as a command line interface (CLI) or Python API. If you use the CLI, export your credentials at the command line and configure `curator.yml` as follows:

```
client:
  hosts: search-my-domain.us-west-1.es.amazonaws.com
  port: 443
  use_ssl: True
  aws_region: us-west-1
  aws_sign_request: True
  ssl_no_validate: False
  timeout: 60

logging:
  loglevel: INFO
```

For sample Lambda functions that use the Python API, see *Using Curator to Rotate Data* (p. 138).

Topics

- Taking Manual Snapshots (p. 91)
- Restoring Snapshots (p. 92)

Manual Snapshot Prerequisites

To create index snapshots manually, you must work with IAM and Amazon S3. Verify that you have met the following prerequisites before you attempt to take a snapshot.

<table>
<tr><th>Prerequisite</th><th>Description</th></tr>
<tr><td>S3 bucket</td><td>

Stores manual snapshots for your Amazon ES domain. Make a note of the bucket's name. You need it in two places:

- `Resource` statement of the IAM policy that is attached to your IAM role
- Python client that is used to register a snapshot repository

For more information, see Create a Bucket in the *Amazon Simple Storage Service Getting Started Guide*.

> **Important**
> Do **not** apply an Amazon Glacier lifecycle rule to this bucket. Manual snapshots do not support the Amazon Glacier storage class.

</td></tr>
<tr><td>IAM role</td><td>

Delegates permissions to Amazon Elasticsearch Service. The rest of this document refers to this role as `TheSnapshotRole`.

The trust relationship for the role must specify Amazon Elasticsearch Service in the `Principal` statement, as shown in the following example:

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

The role must have the following policy attached to it:

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::s3-bucket-name"
      ]
    },
    {
      "Action": [
        "s3:GetObject",
```

</td></tr>
</table>

<table>
<tr><th>Prerequisite</th><th>Description</th></tr>
<tr><td></td><td>

```
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::s3-bucket-name/*"
      ]
    }
  ]
}
```

For more information, see Creating Customer Managed Policies and Attaching Managed Policies in the *IAM User Guide*.
</td></tr>
<tr><td>Permissions</td><td>

You must be able to assume the IAM role in order to register the snapshot repository. You also need access to the `es:ESHttpPut` action. A common way to provide access is to attach the following policy to your account:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/TheSnapshotRole"
    },
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPut",
      "Resource": "arn:aws:es:region:123456789012:domain/my-domain/*"
    }
  ]
}
```

If your account does not have `iam:PassRole` permissions to assume `TheSnapshotRole`, you might encounter the following common error:

```
$ python register-repo.py
{"Message":"User: arn:aws:iam::123456789012:user/MyUserAccount
is not authorized to perform: iam:PassRole on resource:
arn:aws:iam::123456789012:role/TheSnapshotRole"}
```
</td></tr>
</table>

Registering a Manual Snapshot Repository

You must register a snapshot repository with Amazon Elasticsearch Service before you can take manual index snapshots. This one-time operation requires that you sign your AWS request with credentials that are allowed to access `TheSnapshotRole`, as described in the section called “Manual Snapshot Prerequisites” (p. 88).

You can't use `curl` to perform this operation, because it doesn't support AWS request signing. Instead, use the sample Python client (p. 90), Postman, or some other method to send a signed request to register the snapshot repository. The request takes the following form:

```
PUT https://elasticsearch-domain.region.es.amazonaws.com/_snapshot/my-snapshot-repo
{
  "type": "s3",
  "settings": {
    "bucket": "s3-bucket-name",
    "region": "region",
    "role_arn": "arn:aws:iam::123456789012:role/TheSnapshotRole"
  }
}
```

Registering a snapshot directory is a one-time operation, but to migrate from one domain to another, you must register the same snapshot repository on the old domain and the new domain.

Important

If the S3 bucket is in the us-east-1 region, you need to use `"endpoint": "s3.amazonaws.com"` instead of `"region": "us-east-1"`.

To enable server-side encryption with S3-managed keys for the snapshot repository, add `"server_side_encryption": true` to the `"settings"` JSON.

If your domain resides within a VPC, your computer must be connected to the VPC in order for the request to successfully register the snapshot repository. Accessing a VPC varies by network configuration, but likely involves connecting to a VPN or corporate network. To check that you can reach the Amazon ES domain, navigate to `https://your-vpc-domain.region.es.amazonaws.com` in a web browser and verify that you receive the default JSON response.

Sample Python Client

Save the following sample Python code as a Python file, such as `register-repo.py`. The client requires the AWS SDK for Python (Boto 3), requests and requests-aws4auth packages. The client contains commented-out examples for other snapshot operations.

Tip

A Java-based code sample is available in Signing HTTP Requests (p. 62).

You must update the following variables in your code: `host`, `region`, `path`, and `payload`.

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = '' # include https:// and trailing /
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Register repository

path = '_snapshot/my-snapshot-repo' # the Elasticsearch API endpoint
url = host + path

payload = {
  "type": "s3",
  "settings": {
    "bucket": "s3-bucket-name",
    "region": "us-west-1",
    "role_arn": "arn:aws:iam::123456789012:role/TheSnapshotRole"
  }
}

headers = {"Content-Type": "application/json"}

r = requests.put(url, auth=awsauth, json=payload, headers=headers)
```

```
print(r.status_code)
print(r.text)

# # Take snapshot
#
# path = '_snapshot/my-snapshot-repo/my-snapshot'
# url = host + path
#
# r = requests.put(url, auth=awsauth)
#
# print(r.text)
#
# # Delete index
#
# path = 'my-index'
# url = host + path
#
# r = requests.delete(url, auth=awsauth)
#
# print(r.text)
#
# # Restore snapshots (all indices)
#
# path = '_snapshot/my-snapshot-repo/my-snapshot/_restore'
# url = host + path
#
# r = requests.post(url, auth=awsauth)
#
# print(r.text)
#
# # Restore snapshot (one index)
#
# path = '_snapshot/my-snapshot-repo/my-snapshot/_restore'
# url = host + path
#
# payload = {"indices": "my-index"}
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
```

Taking Manual Snapshots

Snapshots are not instantaneous; they take some time to complete. While a snapshot is in-progress, you can still index documents and make other requests to the cluster, but new documents (and updates to existing documents) aren't included in the snapshot. The snapshot includes indices as they existed at the moment you initiated the snapshot.

Elasticsearch snapshots are incremental, meaning that they only store data that has changed since the last successful snapshot. This incremental nature means that the difference in disk usage between frequent and infreqent snapshots is often minimal. In other words, taking hourly snapshots for a week (for a total of 168 snapshots) might not use much more disk space than taking a single snapshot at the end of the week. Also, the more frequently you take snapshots, the less time they take to complete. Some Elasticsearch users take snapshots as often as every half hour.

You specify two pieces of information when you create a snapshot:

- Name of your snapshot repository

- Name for the snapshot

The examples in this chapter use curl, a common HTTP client, for convenience and brevity. If your access policies specify IAM users or roles, however, you must sign your snapshot requests. You can use the commented-out examples in the sample Python client (p. 90) to make signed HTTP requests to the same endpoints that the curl commands use.

To manually take a snapshot

- Run the following command to manually take a snapshot:

```
curl -XPUT 'elasticsearch-domain-endpoint/_snapshot/repository/snapshot-name'
```

Note
The time required to take a snapshot increases with the size of the Amazon ES domain. Long-running snapshot operations commonly encounter the following error: `504 GATEWAY_TIMEOUT`. Typically, you can ignore these errors and wait for the operation to complete successfully. Use the following command to verify the state of all snapshots of your domain:

```
curl -XGET 'elasticsearch-domain-endpoint/_snapshot/repository/_all?pretty'
```

Restoring Snapshots

Warning
If you use index aliases, cease write requests to an alias (or switch the alias to another index) prior to deleting its index. Halting write requests helps avoid the following scenario:

1. You delete an index, which also deletes its alias.
2. An errant write request to the now-deleted alias creates a new index with the same name as the alias.
3. You can no longer use the alias due to a naming conflict with the new index.

If you switched the alias to another index, specify `"include_aliases": false` when you restore from a snapshot.

To restore a snapshot

1. Identify the snapshot that you want to restore. To see all snapshot repositories, run the following command:

```
curl -XGET 'elasticsearch-domain-endpoint/_snapshot?pretty'
```

After you identify the repository, run the following command to see all snapshots:

```
curl -XGET 'elasticsearch-domain-endpoint/_snapshot/repository/_all?pretty'
```

Note
Most automated snapshots are stored in the `cs-automated` repository. If your domain encrypts data at rest, they are stored in the `cs-automated-enc` repository. If you don't

see the manual snapshot repository that you're looking for, make sure that you registered it (p. 89) to the domain.

2. Delete or rename all open indices in the Amazon ES domain.

   You can't restore a snapshot of your indices to an Elasticsearch cluster that already contains indices with the same names. Currently, Amazon ES does not support the Elasticsearch `_close` API, so you must use one of the following alternatives:

   - Delete the indices on the same Amazon ES domain, and then restore the snapshot.
   - Rename the indices as you restore them from the snapshot (p. 147), and later, reindex them.
   - Restore the snapshot to a different Amazon ES domain (only possible with manual snapshots).

   The following example shows how to delete *all* existing indices for a domain:

   ```
   curl -XDELETE 'elasticsearch-domain-endpoint/_all'
   ```

   If you don't plan to restore all indices, though, you might want to delete only one:

   ```
   curl -XDELETE 'elasticsearch-domain-endpoint/index-name'
   ```

3. To restore a snapshot, run the following command:

   ```
   curl -XPOST 'elasticsearch-domain-endpoint/_snapshot/repository/snapshot/_restore'
   ```

   Due to special permissions on the `.kibana` index, attempts to restore all indices might fail, especially if you try to restore from an automated snapshot. The following example restores just one index, `my-index`, from `2017-snapshot` in the `cs-automated` snapshot repository:

   ```
   curl -XPOST 'elasticsearch-domain-endpoint/_snapshot/cs-automated/2017-snapshot/
   _restore' -d '{"indices": "my-index"}' -H 'Content-Type: application/json'
   ```

   Note
   If not all primary shards were available for the indices involved, a snapshot might have a `state` of `PARTIAL`. This value indicates that data from at least one shard was not stored successfully. You can still restore from a partial snapshot, but you might need to use older snapshots to restore any missing indices.

Upgrading Elasticsearch

Amazon ES offers in-place Elasticsearch upgrades for domains that run versions 5.1 and later. If you use services like Amazon Kinesis Data Firehose or Amazon CloudWatch Logs to stream data to Amazon ES, check that these services support the newer version of Elasticsearch before migrating.

Currently, Amazon ES supports the following upgrade paths.

From Version	To Version
6.*x*	6.3
5.6	6.3 **Important** Indices created in version 6.*x* no longer support multiple mapping types. Indices created in version 5.*x* still support multiple mapping types when restored into a 6.*x* cluster. If you use AWS Lambda, check that your code creates only a single mapping type per index. To minimize downtime during the upgrade from Elasticsearch 5.6 to 6.*x*, Amazon ES reindexes the `.kibana` index to `.kibana-6`, deletes `.kibana`, creates an alias named `.kibana`, and maps the new index to the new alias.
5.*x*	5.6

In essence, you can move to the latest release within the same major version (for example, 5.3 to 5.6) or from the latest release in a major version to the latest release in the *next* major version (for example, 5.6 to 6.3). As new Elasticsearch versions become available on Amazon ES, these upgrade paths change.

The upgrade process consists of three steps:

1. **Pre-upgrade checks** – Amazon ES performs a series of checks for issues that can block an upgrade and doesn't proceed to the next step unless these checks succeed.
2. **Snapshot** – Amazon ES takes a snapshot of the Elasticsearch cluster and doesn't proceed to the next step unless the snapshot succeeds. If the upgrade fails, Amazon ES uses this snapshot to restore the cluster to its original state.
3. **Upgrade** – Amazon ES starts the upgrade, which can take from 15 minutes to several hours to complete. Kibana might be unavailable during some or all of the upgrade.

Troubleshooting an Upgrade

In-place Elasticsearch upgrades require healthy domains. Your domain might be ineligible for an upgrade or fail to upgrade for a wide variety of reasons. The following table shows the most common issues.

Issue	Description
Domain in processing	The domain is in the middle of a configuration change. Check upgrade eligibility after the operation completes.
Red cluster status	One or more indices in the cluster is red. For troubleshooting steps, see the section called “Red Cluster Status” (p. 143).

Issue	Description
High error rate	The Elasticsearch cluster is returning a large number of 5*xx* errors when attempting to process requests. This problem is usually the result of too many simultaneous read or write requests. Consider reducing traffic to the cluster or scaling your domain.
Split brain	*Split brain* means that your Elasticsearch cluster has more than one master node and has split into two clusters that never will rejoin on their own. You can avoid split brain by using the recommended number of dedicated master nodes (p. 128). For help recovering from split brain, contact AWS Support.
Master node not found	Amazon ES can't find the cluster's master node. If your domain uses zone awareness, an Availability Zone failure might have caused the cluster to lose quorum and be unable to elect a new master node (p. 128). If the issue does not self-resolve, contact AWS Support.
Too many pending tasks	The master node is under heavy load and has many pending tasks. Consider reducing traffic to the cluster or scaling your domain.
Impaired storage volume	The disk volume of one or more nodes isn't functioning properly. This issue often occurs alongside other issues, like a high error rate or too many pending tasks. If it occurs in isolation and doesn't self-resolve, contact AWS Support.
KMS key issue	The KMS key that is used to encrypt the domain is either inaccessible or missing. For more information, see the section called "Monitoring Domains That Encrypt Data at Rest" (p. 135).
Snapshot in progress	The domain is currently taking a snapshot. Check upgrade eligibility after the snapshot finishes. Also check that you can list manual snapshot repositories, list snapshots within those repositories, and take manual snapshots. If Amazon ES is unable to check whether a snapshot is in progress, upgrades can fail.
Snapshot timeout or failure	The pre-upgrade snapshot took too long to complete or failed. Check cluster health, and try again. If the problem persists, contact AWS Support.
Incompatible indices	One or more indices is incompatible with the target Elasticsearch version. This problem can occur if you migrated the indices from an older version of Elasticsearch, like 2.3. Reindex the indices, and try again.
High disk usage	Disk usage for the cluster is above 90%. Delete data or scale the domain, and try again.
High JVM usage	JVM memory pressure is above 75%. Reduce traffic to the cluster or scale the domain, and try again.
Kibana alias problem	`.kibana` is already configured as an alias and maps to an incompatible index, likely one from an earlier version of Kibana. Reindex, and try again.
Red Kibana status	Kibana status is red. Try using Kibana when the upgrade completes. If the red status persists, resolve it manually, and try again.
Other Amazon ES service issue	Issues with Amazon ES itself might cause your domain to display as ineligible for an upgrade. If none of the preceding conditions apply to your domain and the problem persists for more than a day, contact AWS Support.

Starting an Upgrade

The upgrade process is irreversible and can't be paused nor canceled. During an upgrade, you can't make configuration changes to the domain. Before starting an upgrade, double-check that you want to proceed. You can use these same steps to perform the pre-upgrade check without actually starting an upgrade.

To upgrade a domain to a later version of Elasticsearch (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to upgrade.
4. Choose **Upgrade domain**.
5. For **Operation**, choose **Upgrade**, **Submit**, and **Continue**.
6. Return to the **Overview** tab and choose **Upgrade status** to monitor the state of the upgrade.

To upgrade a domain to a later version of Elasticsearch (AWS CLI and SDK)

You can use the following operations to identify the right Elasticsearch version for your domain, start an in-place upgrade, perform the pre-upgrade check, and view progress:

- `get-compatible-elasticsearch-versions` (`GetCompatibleElasticsearchVersions`)
- `upgrade-elasticsearch-domain` (`UpgradeElasticsearchDomain`)
- `get-upgrade-status` (`GetUpgradeStatus`)
- `get-upgrade-history` (`GetUpgradeHistory`)

For more information, see the AWS CLI Command Reference and *Amazon ES Configuration API Reference* (p. 167).

Using a Snapshot to Migrate Data

In-place upgrades are the easier, faster, and more reliable way to upgrade a domain to a later Elasticsearch version. Snapshots are a good option if you need to migrate from a pre-5.1 version of Elasticsearch or want to migrate to an entirely new cluster.

The following table shows how to use snapshots to migrate data to a domain that uses a different Elasticsearch version. Most of the steps require you to create and restore manual index snapshots. For more information about this process, see *Working with Index Snapshots* (p. 87).

From Version	To Version	Migration Process
6.*x*	6.3	1. Create a manual snapshot of the 6.*x* domain. 2. Create a 6.3 domain. 3. Restore the snapshot from the original domain to the 6.3 domain. 4. If you no longer need your original domain, delete it. Otherwise, you continue to incur charges for the domain.
5.*x*	6.*x*	1. Review breaking changes for 6.0 to see if you need to make adjustments to your indices or application.

From Version	To Version	Migration Process
		For other considerations, see the table in *Upgrading Elasticsearch* (p. 94). 2. Create a manual snapshot of the 5.*x* domain. 3. Create a 6.*x* domain. 4. Restore the snapshot from the original domain to the 6.*x* domain. 5. If you no longer need your 5.*x* domain, delete it. Otherwise, you continue to incur charges for the domain.
5.*x*	5.6	1. Create a manual snapshot of the 5.*x* domain. 2. Create a 5.6 domain. 3. Restore the snapshot from the original domain to the 5.6 domain. 4. If you no longer need your original domain, delete it. Otherwise, you continue to incur charges for the domain.
2.3	6.*x*	Elasticsearch 2.3 snapshots are not compatible with 6.*x*. To migrate your data directly from 2.3 to 6.*x*, you must manually recreate your indices in the new domain. Alternately, you can follow the 2.3 to 5.*x* steps in this table, perform `_reindex` operations in the new 5.*x* domain to convert your 2.3 indices to 5.*x* indices, and then follow the 5.*x* to 6.*x* steps.
2.3	5.*x*	1. Review breaking changes for 5.0 to see if you need to make adjustments to your indices or application. **Note** The Elasticsearch migration plugin currently is not available. 2. Create a manual snapshot of the 2.3 domain. 3. Create a 5.*x* domain. 4. Restore the snapshot from the 2.3 domain to the 5.*x* domain. 5. If you no longer need your 2.3 domain, delete it. Otherwise, you continue to incur charges for the domain.
1.5	5.*x*	Elasticsearch 1.5 snapshots are not compatible with 5.*x*. To migrate your data from 1.5 to 5.*x*, you must manually recreate your indices in the new domain. **Important** 1.5 snapshots *are* compatible with 2.3, but Amazon ES 2.3 domains do not support the `_reindex` operation. Because you cannot reindex them, indices that originated in a 1.5 domain still fail to restore from 2.3 snapshots to 5.*x* domains.

From Version	To Version	Migration Process
1.5	2.3	1. Use the `_plugin/migration` Elasticsearch plugin to find out if you can directly upgrade to version 2.3. You might need to make changes to your data before migration. a. In a web browser, open `http://`*`domain_endpoint`*`/_plugin/migration/`. b. Choose **Run checks now**. c. Review the results and, if needed, follow the instructions to make changes to your data. 2. Create a manual snapshot of the 1.5 domain. 3. Create a 2.3 domain. 4. Restore the snapshot from the 1.5 domain to the 2.3 domain. 5. If you no longer need your 1.5 domain, delete it. Otherwise, you continue to incur charges for the domain.

Kibana and Logstash

This chapter describes some considerations for using Kibana and Logstash with Amazon Elasticsearch Service.

Topics

Kibana

Kibana is a popular open source visualization tool designed to work with Elasticsearch. Amazon ES provides an installation of Kibana with every Amazon ES domain. You can find a link to Kibana on your domain dashboard on the Amazon ES console. The URL is `https://domain.region.es.amazonaws.com/_plugin/kibana/`. Queries using this default Kibana installation have a 60-second timeout.

The following sections address some common Kibana use cases:

- the section called “Controlling Access to Kibana” (p. 99)
- the section called “Configuring Kibana to Use a WMS Map Server” (p. 101)
- the section called “Connecting a Local Kibana Server to Amazon ES” (p. 102)

Controlling Access to Kibana

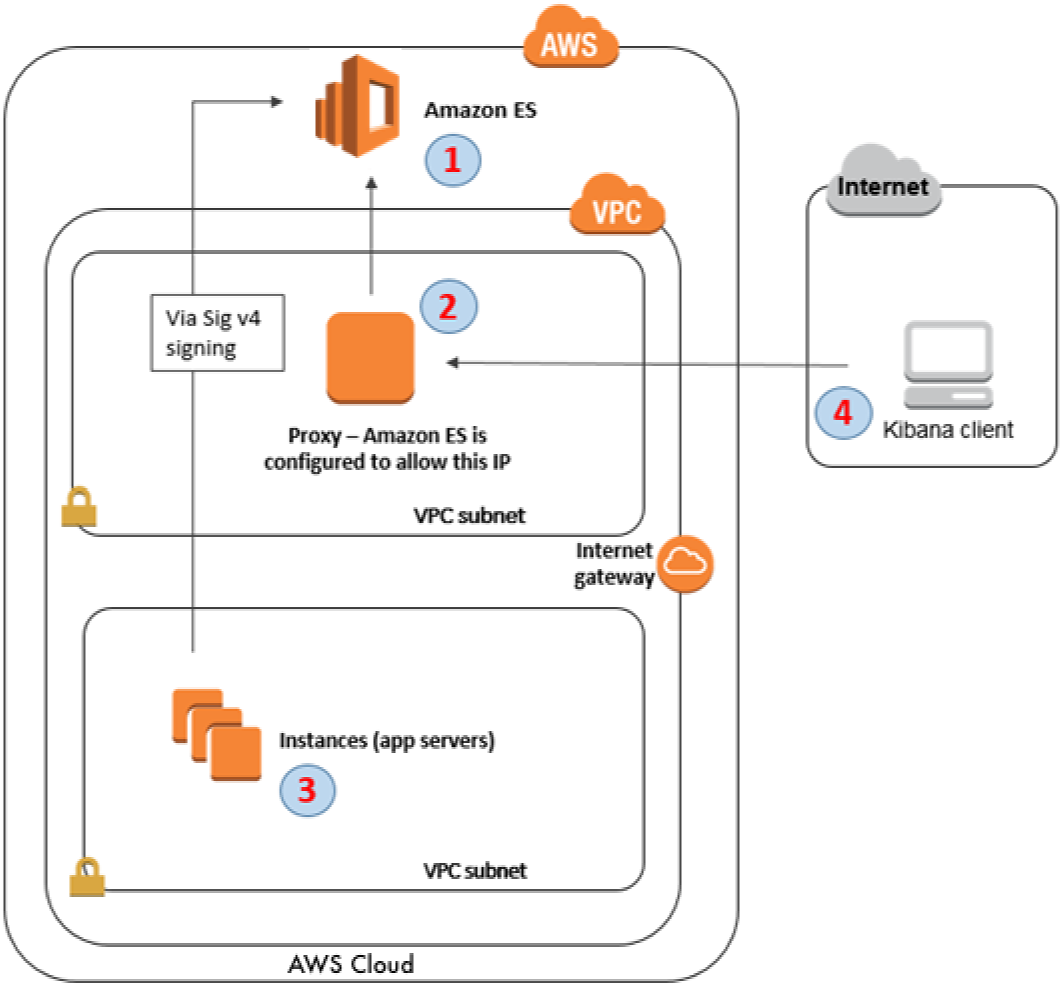
Kibana does not natively support IAM users and roles, but Amazon ES offers several solutions for controlling access to Kibana:

Domain Configuration	Access Control Options
Public access	• Configure *Authentication for Kibana* (p. 104). • Configure an IP-based access policy (p. 33), with or without a proxy server (p. 99).
VPC access	• Configure *Authentication for Kibana* (p. 104). • Configure an open access policy, with or without a proxy server, and use security groups to control access. To learn more, see the section called “About Access Policies on VPC Domains” (p. 119).

Using a Proxy to Access Amazon ES from Kibana

Note
This process is only applicable if your domain uses public access and you don't want to use *Authentication for Kibana* (p. 104). See the section called “Controlling Access to Kibana” (p. 99).

Because Kibana is a JavaScript application, requests originate from the user's IP address. IP-based access control might be impractical due to the sheer number of IP addresses you would need to whitelist in order for each user to have access to Kibana. One workaround is to place a proxy server between Kibana and Amazon ES. Then you can add an IP-based access policy that allows requests from only one IP address, the proxy's. The following diagram shows this configuration.



1. This is your Amazon ES domain. IAM provides authorized access to this domain. An additional, IP-based access policy provides access to the proxy server.
2. This is the proxy server, running on an Amazon EC2 instance.
3. Other applications can use the Signature Version 4 signing process to send authenticated requests to Amazon ES.
4. Kibana clients connect to your Amazon ES domain through the proxy.

To enable this sort of configuration, you need a resource-based policy that specifies roles and IP addresses. Here's a sample policy:

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Resource": "arn:aws:es:us-west-2:111111111111:domain/my-domain/*",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:role/allowedrole1"
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "123.456.789.123"
          ]
        }
      },
      "Resource": "arn:aws:es:us-west-2:111111111111:domain/my-domain/*"
    }
  ]
}
```

We recommend that you configure the EC2 instance running the proxy server with an Elastic IP address. This way, you can replace the instance when necessary and still attach the same public IP address to it. To learn more, see Elastic IP Addresses in the *Amazon EC2 User Guide for Linux Instances*.

If you use a proxy server *and* *Authentication for Kibana* (p. 104), you might need to add settings for Kibana and Amazon Cognito to avoid `redirect_mismatch` errors. See the following `nginx.conf` example:

```
server {
  listen 443;

  location /login {
    proxy_pass           https://$cognito_host/login;
    proxy_cookie_domain  $cognito_host                 $proxy_host;
    proxy_redirect       https://$kibana_host          https://$proxy_host;
  }

  location / {
    proxy_pass               https://$kibana_host;
    proxy_redirect           https://$cognito_host  https://$proxy_host;
    proxy_cookie_domain      $kibana_host           $proxy_host;
    proxy_buffer_size        128k;
    proxy_buffers            4                      256k;
    proxy_busy_buffers_size  256k;
  }
}

$cognito_host=your-cognito-domain-name.auth.us-west-2.amazoncognito.com
$kibana_host=search-your-es-domain.us-west-2.es.amazonaws.com
$proxy_host=your-proxy-server.us-west-2.compute.amazonaws.com
```

Configuring Kibana to Use a WMS Map Server

Due to licensing restrictions, the default installation of Kibana on Amazon ES domains that use Elasticsearch 5.*x* or greater does *not* include a map server for tile map visualizations. Use the following procedure to configure Kibana to use a Web Map Service (WMS) map server.

To configure Kibana to use a WMS map server:

1. Open Kibana. You can find a link to Kibana in the domain summary at https://console.aws.amazon.com/es/.

2. Choose **Management**.
3. Choose **Advanced Settings**.
4. Locate **visualization:tileMap:WMSdefaults**, and then choose the **edit** button to modify the default value.
5. Change `enabled` to `true` and `url` to the URL of a valid WMS map server.
6. (Optional) Locate **visualization:tileMap:WMSdefaults**, and then choose the **edit** button to modify the default value.
7. (Optional) Change `"layers": "0"` to a comma-separated list of map layers that you want to display. Layers vary by map service. The default value of `0` is often appropriate.
8. Choose the **save** button.

To apply the new default value to visualizations, you might need to reload Kibana.

> **Note**
> Map services often have licensing fees or restrictions. You are responsible for all such considerations on any map server that you specify. You might find the map services from the U.S. Geological Survey useful for testing.

Connecting a Local Kibana Server to Amazon ES

If you have invested significant time into configuring your own Kibana instance, you can use it instead of (or in addition to) the default Kibana instance that Amazon ES provides.

To connect a local Kibana server to Amazon ES:

- Make the following changes to `config/kibana.yml`:

```
kibana_index: ".kibana-5"
elasticsearch_url: "http://elasticsearch_domain_endpoint:80"
```

You must use the `http` prefix and explicitly specify port 80.

Loading Bulk Data with the Logstash Plugin

Logstash provides a convenient way to use the bulk API to upload data into your Amazon ES domain with the S3 plugin. The service also supports all other standard Logstash input plugins that are provided by Elasticsearch. Amazon ES also supports two Logstash output plugins: the standard Elasticsearch plugin and the logstash-output-amazon-es plugin, which signs and exports Logstash events to Amazon ES.

You must install your own local instance of Logstash and make the following changes in the Logstash configuration file to enable interaction with Amazon ES.

Configuration Field	Input | Output Plugin	Description
`bucket`	Input	Specifies the Amazon S3 bucket containing the data that you want to load into an Amazon ES domain.
`region`	Input	Specifies the AWS Region where the Amazon S3 bucket resides.

Configuration Field	Input | Output Plugin	Description
`hosts`	Output	Specifies the service endpoint for the target Amazon ES domain.
`ssl`	Output	Specifies whether to use SSL to connect to Amazon ES.

This example configures Logstash to do the following:

- Point the output plugin to an Amazon ES endpoint
- Point to the input plugin to the `wikipedia-stats-log` bucket in S3
- Use SSL to connect to Amazon ES

```
input{
    s3 {
        bucket => "wikipedia-stats-log"
        access_key_id => "lizards"
        secret_access_key => "lollipops"
        region => "us-east-1"
    }
}
output{
    elasticsearch {
        hosts => "search-logs-demo0-cpxczkdpi4bkb4c44g3csyln5a.us-east-1.es.example.com"
        ssl => true
    }
}
```

Note
The service request in the preceding example must be signed. For more information about signing requests, see the section called “Making and Signing Amazon ES Requests” (p. 33). Use the logstash-output-amazon-es output plugin to sign and export Logstash events to Amazon ES. For instructions, see the plugin `README`.

Amazon Cognito Authentication for Kibana

Amazon Elasticsearch Service uses Amazon Cognito to offer user name and password protection for Kibana (p. 99). This authentication feature is optional and available only for domains using Elasticsearch 5.1 or later. If you don't configure Amazon Cognito authentication, you can still protect Kibana using an IP-based access policy (p. 33) and a proxy server (p. 99).

Much of the authentication process occurs in Amazon Cognito, but this chapter offers guidelines and requirements for configuring Amazon Cognito resources to work with Amazon ES domains. Standard pricing applies to all Amazon Cognito resources.

> **Tip**
> The first time that you configure a domain to use Amazon Cognito authentication for Kibana, we recommend using the console. Amazon Cognito resources are extremely customizable, and the console can help you identity and understand the features that matter to you.

Topics

Prerequisites

Before you can configure Amazon Cognito authentication for Kibana, you must fulfill several prerequisites. The Amazon ES console helps streamline the creation of these resources, but understanding the purpose of each resource helps with configuration and troubleshooting. Amazon Cognito authentication for Kibana requires the following resources:

- Amazon Cognito user pool
- Amazon Cognito identity pool
- IAM role that has the `AmazonESCognitoAccess` policy attached

> **Note**
> The user pool and identity pool must be in the same AWS Region. You can use the same user pool, identity pool, and IAM role to add Amazon Cognito authentication for Kibana to multiple Amazon ES domains. To learn more, see the section called “Limits” (p. 113).

About the User Pool

User pools have two main features: create and manage a directory of users, and let users sign up and log in. For instructions about creating a user pool, see Create a User Pool in the *Amazon Cognito Developer Guide*.

When you create a user pool to use with Amazon ES, consider the following:

- Your Amazon Cognito user pool must have a domain name. Amazon ES uses this domain name to redirect users to a login page for accessing Kibana. Other than a domain name, the user pool doesn't require any non-default configuration.
- You must specify the pool's required standard attributes—attributes like name, birth date, email address, and phone number. You can't change these attributes after you create the user pool, so choose the ones that matter to you at this time.
- While creating your user pool, choose whether users can create their own accounts, the minimum password strength for accounts, and whether to enable multi-factor authentication. If you plan to use an external identity provider, these settings are inconsequential. Technically, you can enable the user pool as an identity provider *and* enable an external identity provider, but most people prefer one or the other.

User pool IDs take the form of `region_ID`. If you plan to use the AWS CLI or an AWS SDK to configure Amazon ES, make note of the ID.

About the Identity Pool

Identity pools let you assign temporary, limited-privilege roles to users after they log in. For instructions about creating an identity pool, see Identity Pools in the *Amazon Cognito Developer Guide*. When you create an identity pool to use with Amazon ES, consider the following:

- If you use the Amazon Cognito console, you must select the **Enable access to unauthenticated identities** check box to create the identity pool. After you create the identity pool and configure the Amazon ES domain (p. 106), Amazon Cognito disables this setting.
- You don't need to add external identity providers to the identity pool. When you configure Amazon ES to use Amazon Cognito authentication, it configures the identity pool to use the user pool that you just created.
- After you create the identity pool, you must choose unauthenticated and authenticated IAM roles. These roles specify the access policies that users have before and after they log in. If you use the Amazon Cognito console, it can create these roles for you. After you create the authenticated role, make note of the ARN, which takes the form of `arn:aws:iam::123456789012:role/Cognito_identitypoolAuth_Role`.

Identity pool IDs take the form of `region:ID-ID-ID-ID-ID`. If you plan to use the AWS CLI or an AWS SDK to configure Amazon ES, make note of the ID.

About the IAM Role

Amazon ES needs permissions to configure the Amazon Cognito user and identity pools and use them for authentication. You can use `AmazonESCognitoAccess`, which is an AWS managed policy, for this purpose. If you use the console to create or configure your Amazon ES domain, it creates an IAM role for you and attaches this policy to the role.

If you use the AWS CLI or one of the AWS SDKs, you must create your own role, attach the policy, and specify the ARN for this role when you configure your Amazon ES domain. The role must have the following trust relationship:

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

For instructions, see Creating a Role to Delegate Permissions to an AWS Service and Attaching and Detaching IAM Policies in the *IAM User Guide*.

Configuring an Amazon ES Domain

After you complete the prerequisites, you can configure an Amazon ES domain to use Amazon Cognito for Kibana.

> **Note**
> Amazon Cognito is not available in all AWS Regions. For a list of supported regions, see AWS Regions and Endpoints. You don't need to use the same region for Amazon Cognito that you use for Amazon ES.

Configuring Amazon Cognito Authentication (Console)

Because it creates the IAM role (p. 105) for you, the console offers the simplest configuration experience. In addition to the standard Amazon ES permissions, you need the following set of permissions to use the console to create a domain that uses Amazon Cognito authentication for Kibana:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole",
        "iam:CreateRole",
        "iam:AttachRolePolicy",
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools"
      ],
      "Resource": "*"
    }
  ]
}
```

If the IAM role (p. 105) already exists, you need fewer permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/CognitoAccessForAmazonES"
    }
  ]
}
```

To configure Amazon Cognito authentication for Kibana (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to configure.
4. Choose **Configure cluster**.
5. For **Kibana authentication**, choose **Enable Amazon Cognito for authentication**.
6. For **Region**, select the region that contains your Amazon Cognito user pool and identity pool.
7. For **Cognito User Pool**, select a user pool or create one. For guidance, see the section called “About the User Pool” (p. 105).
8. For **Cognito Identity Pool**, select an identity pool or create one. For guidance, see the section called “About the Identity Pool” (p. 105).

   Note

   The **Create new user pool** and **Create new identity pool** links direct you to the Amazon Cognito console and require you to create these resources manually. The process is not automatic. To learn more, see the section called “Prerequisites” (p. 104).
9. For **IAM Role**, use the default value (recommended) or type a new name. To learn more about the purpose of this role, see the section called “About the IAM Role” (p. 105).
10. Choose **Submit**.

After your domain finishes processing, see the section called “Allowing the Authenticated Role” (p. 108) and the section called “Configuring Identity Providers” (p. 109) for additional configuration steps.

Configuring Amazon Cognito Authentication (AWS CLI)

Use the `--cognito-options` parameter to configure your Amazon ES domain. The following syntax is used by both the `create-elasticsearch-domain` and `update-elasticsearch-domain-config` commands:

```
--cognito-options Enabled=true,UserPoolId="user-pool-id",IdentityPoolId="identity-pool-
id",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonES"
```

Example

The following example creates a domain in the `us-east-1` Region that enables Amazon Cognito authentication for Kibana using the `CognitoAccessForAmazonES` role and provides domain access to `Cognito_Auth_Role`:

```
aws es create-elasticsearch-domain --domain-name my-domain --region us-east-1 --access-
policies '{ "Version":"2012-10-17", "Statement":[{"Effect":"Allow","Principal":{"AWS":
 ["arn:aws:iam::123456789012:role/
Cognito_Auth_Role"]},"Action":"es:ESHttp*","Resource":"arn:aws:es:us-
east-1:123456789012:domain/*" }]}' --elasticsearch-version "6.0" --elasticsearch-
cluster-config InstanceType=m4.xlarge.elasticsearch,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeSize=10 --cognito-options
 Enabled=true,UserPoolId="us-east-1_123456789",IdentityPoolId="us-
east-1:12345678-1234-1234-1234-123456789012",RoleArn="arn:aws:iam::123456789012:role/
CognitoAccessForAmazonES"
```

After your domain finishes processing, see the section called “Allowing the Authenticated Role” (p. 108) and the section called “Configuring Identity Providers” (p. 109) for additional configuration steps.

Configuring Amazon Cognito Authentication (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the operations that are defined in the *Amazon ES Configuration API Reference* (p. 167), including the `CognitoOptions` parameter for the `CreateElasticsearchDomain` and `UpdateElasticsearchDomainConfig` operations. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

After your domain finishes processing, see the section called “Allowing the Authenticated Role” (p. 108) and the section called “Configuring Identity Providers” (p. 109) for additional configuration steps.

Allowing the Authenticated Role

By default, the authenticated IAM role that you configured by following the guidelines in the section called “About the Identity Pool” (p. 105) does not have the necessary privileges to access Kibana. You must provide the role with additional permissions.

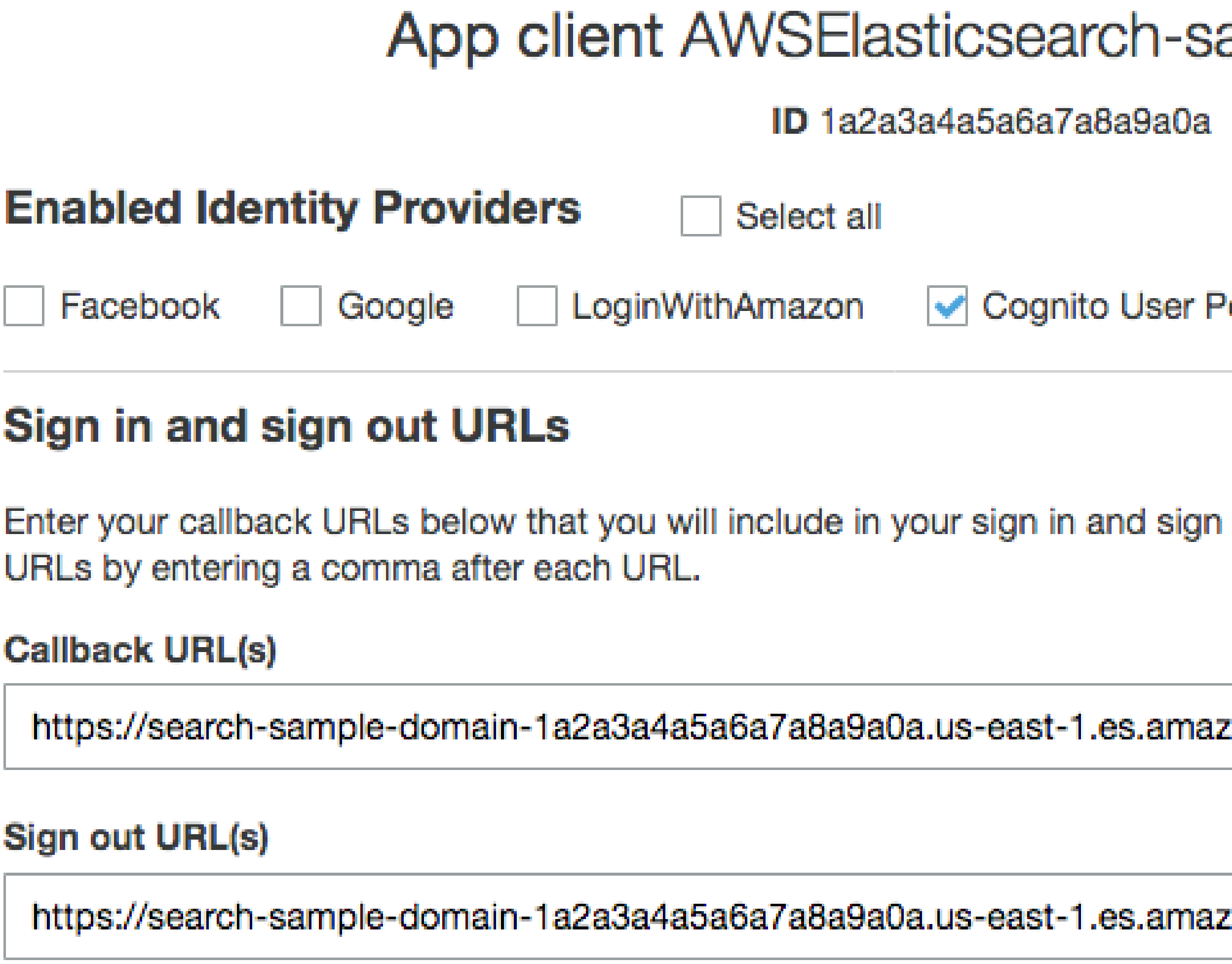
You can include these permissions in an identity-based (p. 32) policy, but unless you want authenticated users to have access to all Amazon ES domains, a resource-based (p. 30) policy attached to a single domain is the more common approach:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/Cognito_identitypoolAuth_Role"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:123456789012:domain/domain-name/*"
    }
  ]
}
```

For instructions about adding a resource-based policy to an Amazon ES domain, see the section called "Configuring Access Policies" (p. 21).

Configuring Identity Providers

When you configure a domain to use Amazon Cognito authentication for Kibana, Amazon ES adds an app client to the user pool and adds the user pool to the identity pool as an authentication provider. The following screenshot shows the **App client settings** page in the Amazon Cognito console.



App client AWSElasticsearch-sa

ID 1a2a3a4a5a6a7a8a9a0a

Enabled Identity Providers □ Select all

□ Facebook □ Google □ LoginWithAmazon ☑ Cognito User P

Sign in and sign out URLs

Enter your callback URLs below that you will include in your sign in and sign URLs by entering a comma after each URL.

Callback URL(s)

https://search-sample-domain-1a2a3a4a5a6a7a8a9a0a.us-east-1.es.amaz

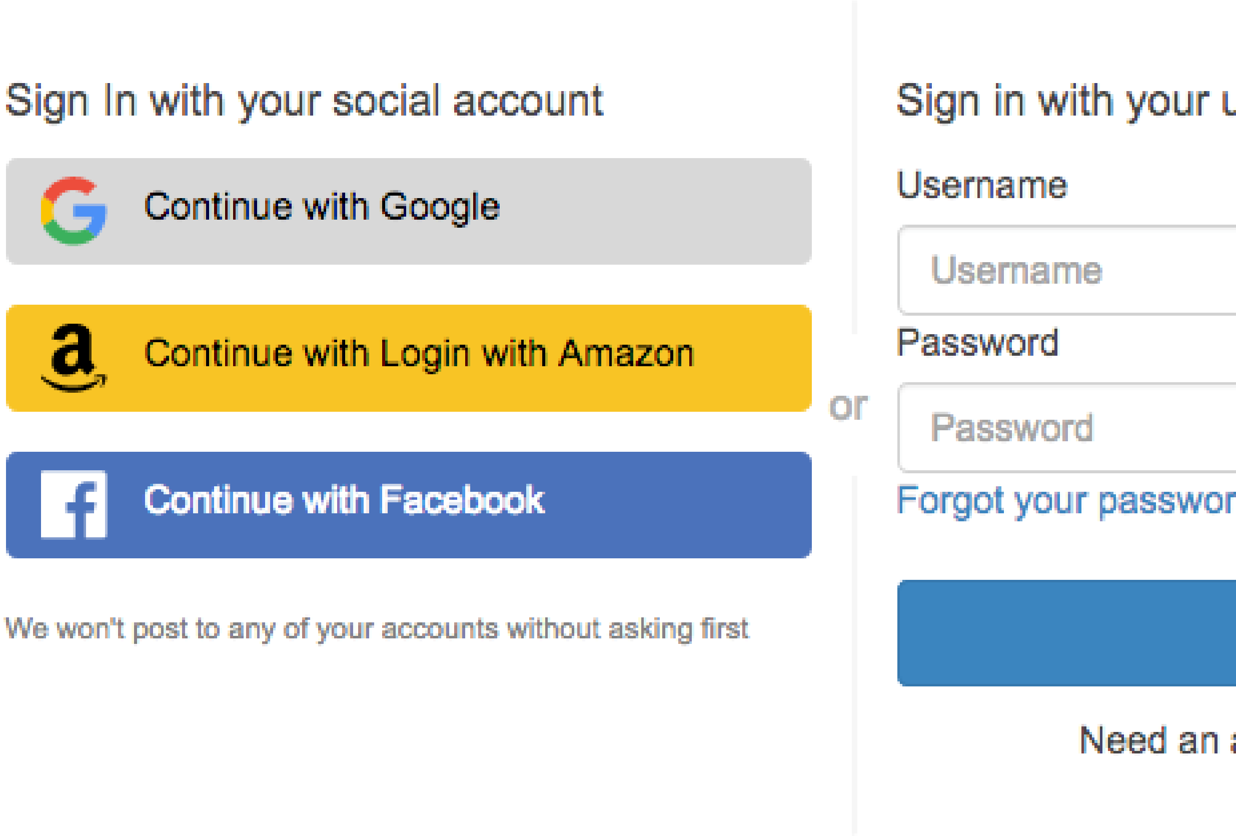
Sign out URL(s)

https://search-sample-domain-1a2a3a4a5a6a7a8a9a0a.us-east-1.es.amaz

Warning
Don't rename or delete the app client.

Depending on how you configured your user pool, you might need to create user accounts manually, or users might be able to create their own. If these settings are acceptable, you don't need to take further action. Many people, however, prefer to use external identity providers.

To enable a SAML 2.0 identity provider, you must provide a SAML metadata document. To enable social identity providers like Login with Amazon, Facebook, and Google, you must have an app ID and app secret from those providers. You can enable any combination of identity providers. The login page adds options as you add providers, as shown in the following screenshot.



Sign In with your social account

We won't post to any of your accounts without asking first

Sign in with your u

Username: Username

Password: Password

The easiest way to configure your user pool is to use the Amazon Cognito console. Use the **Identity Providers** page to add external identity providers and the **App client settings** page to enable and disable identity providers for the Amazon ES domain's app client. For example, you might want to enable your own SAML identity provider and disable **Cognito User Pool** as an identity provider.

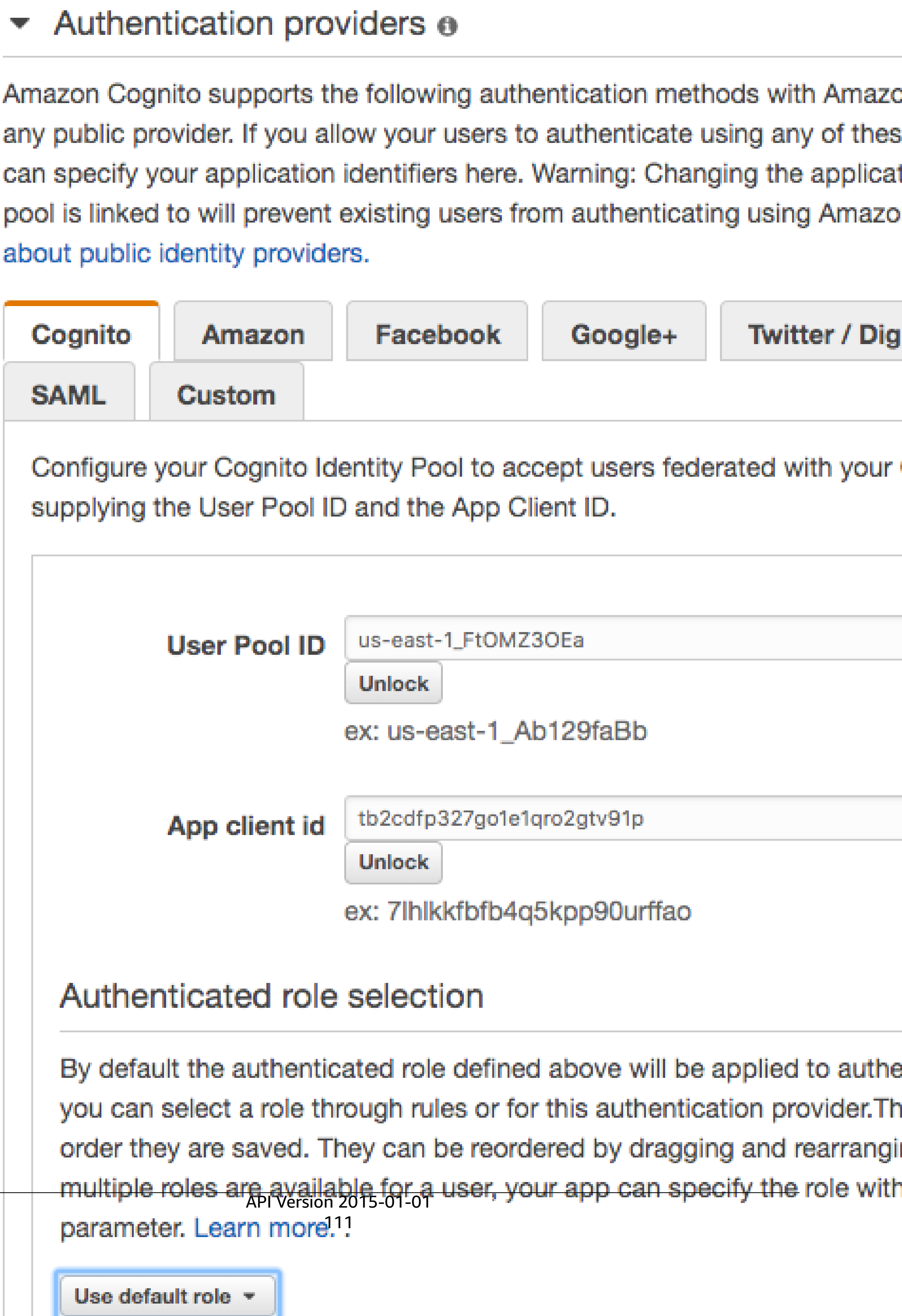
For instructions, see Using Federation from a User Pool and Specifying Identity Provider Settings for Your User Pool App in the *Amazon Cognito Developer Guide*.

(Optional) Configuring Granular Access

You might have noticed that the default identity pool settings assign every user who logs in the same IAM role (`Cognito_`*`identitypool`*`Auth_Role`), which means that every user can access the same AWS resources. If you want more granular access control—for example, if you want your organization's analysts to have access to all eight of your Amazon ES domains, but everyone else to have access to only five of them—you have two options:

- Create user groups and configure your identity provider to choose the IAM role based on the user's authentication token.
- Configure your identity provider to choose the IAM role based on one or more rules.

You configure these options using the **Edit identity pool** page of the Amazon Cognito console, as shown in the following screenshot.



Amazon Cognito supports the following authentication methods with Amazo any public provider. If you allow your users to authenticate using any of thes can specify your application identifiers here. Warning: Changing the applicat pool is linked to will prevent existing users from authenticating using Amazo about public identity providers.

Configure your Cognito Identity Pool to accept users federated with your supplying the User Pool ID and the App Client ID.

User Pool ID: us-east-1_FtOMZ3OEa

ex: us-east-1_Ab129faBb

App client id: tb2cdfp327go1e1qro2gtv91p

ex: 7lhlkkfbfb4q5kpp90urffao

Authenticated role selection

By default the authenticated role defined above will be applied to authe you can select a role through rules or for this authentication provider.Th order they are saved. They can be reordered by dragging and rearrangi multiple roles are available for a user, your app can specify the role with parameter. Learn more.

User Groups and Tokens

When you create a user group, you choose an IAM role for members of the group. For information about creating groups, see User Groups in the *Amazon Cognito Developer Guide*.

After you create one or more user groups, you can configure your authentication provider to assign users their groups' roles rather than the identity pool's default role. Choose the **Choose role from token** option. Then choose either **Use default Authenticated role** or **DENY** to specify how the identity pool should handle users who are not part of a group.

Rules

Rules are essentially a series of `if` statements that Amazon Cognito evaluates sequentially. For example, if a user's email address contains `@corporate`, Amazon Cognito assigns that user `Role_A`. If a user's email address contains `@subsidiary`, it assigns that user `Role_B`. Otherwise, it assigns the user the default authenticated role.

To learn more, see Using Rule-Based Mapping to Assign Roles to Users in the *Amazon Cognito Developer Guide*.

(Optional) Customizing the Login Page

The **UI customization** page of the Amazon Cognito console lets you upload a custom logo and make CSS changes to the login page. For instructions and a full list of CSS properties, see Specifying App UI Customization Settings for Your User Pool in the *Amazon Cognito Developer Guide*.

(Optional) Configuring Advanced Security

Amazon Cognito user pools support advanced security features like multi-factor authentication, compromised credential checking, and adaptive authentication. To learn more, see Managing Security in the *Amazon Cognito Developer Guide*.

Testing

After you are satisfied with your configuration, verify that the user experience meets your expectations.

To access Kibana

1. Navigate to `https://elasticsearch-domain/_plugin/kibana/` in a web browser.
2. Log in using your preferred credentials.
3. After Kibana loads, configure at least one index pattern. Kibana uses these patterns to identity which indices that you want to analyze. Enter `*`, choose **Next step**, and then choose **Create index pattern**.
4. To search or explore your data, choose **Discover**.

If any step of this process fails, see the section called “Common Configuration Issues” (p. 113) for troubleshooting information.

Limits

Amazon Cognito has soft limits on many of its resources. If you want to enable Kibana authentication for a large number of Amazon ES domains, review Limits in Amazon Cognito and request limit increases as necessary.

Each Amazon ES domain adds an app client to the user pool, which adds an authentication provider to the identity pool. If you enable Kibana authentication for more than 10 domains, you might encounter the "maximum Amazon Cognito user pool providers per identity pool" limit. If you exceed a limit, any Amazon ES domains that you try to configure to use Amazon Cognito authentication for Kibana can get stuck in a configuration state of **Processing**.

Common Configuration Issues

The following tables list common configuration issues and solutions.

Configuring Amazon ES

<table>
<tr><th>Issue</th><th>Solution</th></tr>
<tr><td><code>Amazon ES can't create the role</code> (console)</td><td>You don't have the correct IAM permissions. Add the permissions specified in the section called “Configuring Amazon Cognito Authentication (Console)” (p. 106).</td></tr>
<tr><td><code>User is not authorized to perform: iam:PassRole on resource CognitoAccessForAmazonES</code> (console)</td><td>You don't have <code>iam:PassRole</code> permissions for the IAM role (p. 105). Attach the following policy to your account:
<pre>
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/
service-role/CognitoAccessForAmazonES"
    }
  ]
}
</pre>
Alternately, you can attach the <code>IAMFullAccess</code> policy.</td></tr>
<tr><td><code>User is not authorized to perform: cognito-identity:ListIdentityPools on resource</code></td><td>You don't have read permissions for Amazon Cognito. Attach the <code>AmazonCognitoReadOnly</code> policy to your account.</td></tr>
<tr><td><code>An error occurred (ValidationException) when calling the CreateElasticsearchDomain operation: Amazon Elasticsearch must be allowed to use the passed role</code></td><td>Amazon ES isn't specified in the trust relationship of the IAM role. Check that your role uses the trust relationship that is specified in the section called “About the IAM Role” (p. 105). Alternately, use the console to configure Amazon Cognito authentication. The console creates a role for you.</td></tr>
</table>

Issue	Solution
`An error occurred (ValidationException) when calling the CreateElasticsearchDomain operation: User is not authorized to perform: cognito-idp:action on resource: user pool`	The role specified in `--cognito-options` does not have permissions to access Amazon Cognito. Check that the role has the AWS managed `AmazonESCognitoAccess` policy attached. Alternately, use the console to configure Amazon Cognito authentication. The console creates a role for you.
`An error occurred (ValidationException) when calling the CreateElasticsearchDomain operation: User pool does not exist`	Amazon ES can't find the user pool. Confirm that you created one and have the correct ID. To find the ID, you can use the Amazon Cognito console or the following AWS CLI command: `aws cognito-idp list-user-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateElasticsearchDomain operation: IdentityPool not found`	Amazon ES can't find the identity pool. Confirm that you created one and have the correct ID. To find the ID, you can use the Amazon Cognito console or the following AWS CLI command: `aws cognito-identity list-identity-pools --max-results 60 --region region`
`An error occurred (ValidationException) when calling the CreateElasticsearchDomain operation: Domain needs to be specified for user pool`	The user pool does not have a domain name. You can configure one using the Amazon Cognito console or the following AWS CLI command: `aws cognito-idp create-user-pool-domain --domain name --user-pool-id id`

Accessing Kibana

Issue	Solution
The login page doesn't show my preferred identity providers.	Check that you enabled the identity provider for the Amazon ES app client as specified in the section called “Configuring Identity Providers” (p. 109).
The login page doesn't look as if it's associated with my organization.	See the section called “(Optional) Customizing the Login Page” (p. 112).
My login credentials don't work.	Check that you have configured the identity provider as specified in the section called “Configuring Identity Providers” (p. 109). If you use the user pool as your identity provider, check that the account exists and is confirmed on the **User and groups** page of the Amazon Cognito console.
Kibana either doesn't load at all or doesn't work properly.	The Amazon Cognito authenticated role needs `es:ESHttp*` permissions for the domain (`/*`) to access and use Kibana.

<table>
<tr><th>Issue</th><th>Solution</th></tr>
<tr><td></td><td>Check that you added an access policy as specified in the section called “Allowing the Authenticated Role” (p. 108).</td></tr>
<tr><td><code>Invalid identity pool configuration. Check assigned IAM roles for this pool.</code></td><td>Amazon Cognito can't assume the authenticated role. If you used a preexisting role rather than creating a new one for the identity pool, modify the trust relationship for the authenticated role:
<pre>
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "cognito-identity.amazonaws.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity"
    }
  ]
}
</pre>
Alternately, you can create a new role using the Amazon Cognito console.</td></tr>
<tr><td><code>Token is not from a supported provider of this identity pool.</code></td><td>This uncommon error can occur when you remove the app client from the user pool. Try opening Kibana in a new browser session.</td></tr>
</table>

Disabling Amazon Cognito Authentication for Kibana

Use the following procedure to disable Amazon Cognito authentication for Kibana.

To disable Amazon Cognito authentication for Kibana (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. In the navigation pane, under **My domains**, choose the domain that you want to configure.
4. Choose **Configure cluster**.
5. For **Kibana authentication**, clear the **Enable Amazon Cognito for authentication** check box.
6. Choose **Submit**.

> **Important**
> If you no longer need the Amazon Cognito user pool and identity pool, delete them. Otherwise, you can continue to incur charges.

Deleting Domains that Use Amazon Cognito Authentication for Kibana

To prevent domains that use Amazon Cognito authentication for Kibana from becoming stuck in a configuration state of **Processing**, delete Amazon ES domains *before* deleting their associated Amazon Cognito user pools and identity pools.

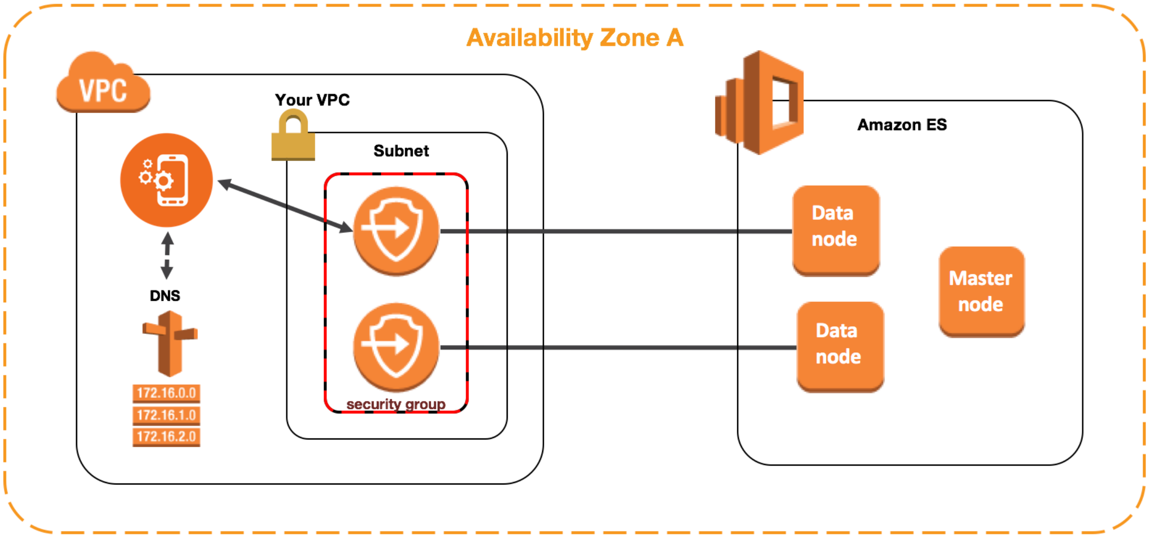
VPC Support for Amazon Elasticsearch Service Domains

A *virtual private cloud* (VPC) is a virtual network that is dedicated to your AWS account. It's logically isolated from other virtual networks in the AWS Cloud. You can launch AWS resources, such as Amazon ES domains, into your VPC.

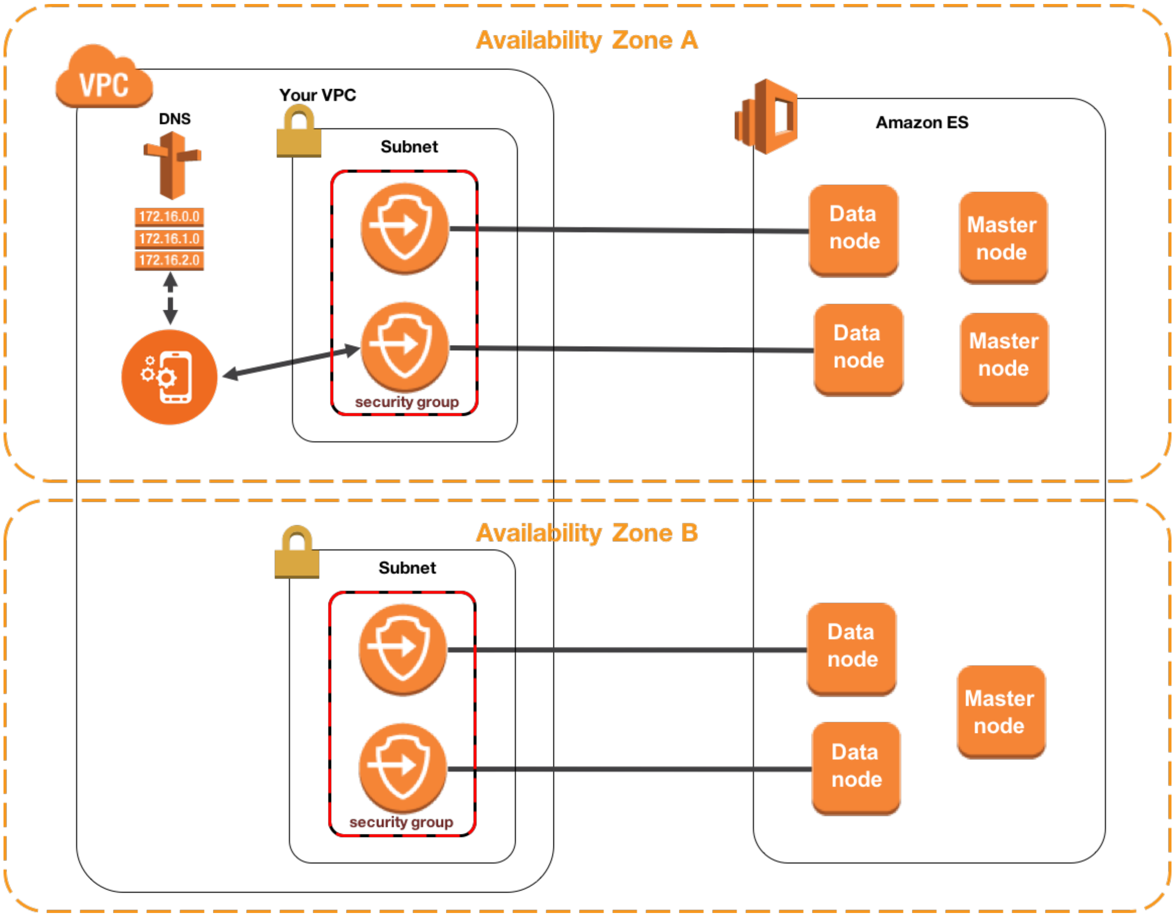
Placing an Amazon ES domain within a VPC enables secure communication between Amazon ES and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection. All traffic remains securely within the AWS Cloud. Because of their logical isolation, domains that reside within a VPC have an extra layer of security when compared to domains that use public endpoints.

To support VPCs, Amazon ES places an endpoint into either one or two subnets of your VPC. A *subnet* is a range of IP addresses in your VPC. If you enable zone awareness (p. 43) for your domain, Amazon ES places an endpoint into two subnets. The subnets must be in different Availability Zones in the same region. If you don't enable zone awareness, Amazon ES places an endpoint into only one subnet.

The following illustration shows the VPC architecture if zone awareness is not enabled.



The following illustration shows the VPC architecture if zone awareness is enabled.



Amazon ES also places an *elastic network interface* (ENI) in the VPC for each of your data nodes. Amazon ES assigns each ENI a private IP address from the IPv4 address range of your subnet. The service also assigns a public DNS hostname (which is the domain endpoint) for the IP addresses. You must use a public DNS service to resolve the endpoint (which is a DNS hostname) to the appropriate IP addresses for the data nodes:

- If your VPC uses the Amazon-provided DNS server by setting the `enableDnsSupport` option to `true` (the default value), resolution for the Amazon ES endpoint will succeed.
- If your VPC uses a private DNS server and the server can reach the public authoritative DNS servers to resolve DNS hostnames, resolution for the Amazon ES endpoint will also succeed.

Because the IP addresses might change, you should resolve the domain endpoint periodically so that you can always access the correct data nodes. We recommend that you set the DNS resolution interval to one minute. If you're using a client, you should also ensure that the DNS cache in the client is cleared.

> **Note**
> Amazon ES doesn't support IPv6 addresses with a VPC. You can use a VPC that has IPv6 enabled, but the domain will use IPv4 addresses.

Topics

Limitations

Currently, operating an Amazon ES domain within a VPC has the following limitations:

- You can either launch your domain within a VPC or use a public endpoint, but you can't do both. You must choose one or the other when you create your domain.
- If you launch a new domain within a VPC, you can't later switch it to use a public endpoint. The reverse is also true: If you create a domain with a public endpoint, you can't later place it within a VPC. Instead, you must create a new domain and migrate your data.
- You can't launch your domain within a VPC that uses dedicated tenancy. You must use a VPC with tenancy set to **Default**.
- After you place a domain within a VPC, you can't move it to a different VPC. However, you can change the subnets and security group settings.
- Compared to public domains, VPC domains display less information in the Amazon ES console. Specifically, the **Cluster health** tab does not include shard information, and the **Indices** tab is not present at all.
- Currently, Amazon ES does not support integration with Amazon Kinesis Data Firehose for domains that reside within a VPC. To use this service with Amazon ES, you must use a domain with public access.
- To access the default installation of Kibana for a domain that resides within a VPC, users must have access to the VPC. This process varies by network configuration, but likely involves connecting to a VPN or managed network or using a proxy server. To learn more, see the section called "About Access Policies on VPC Domains" (p. 119), the Amazon VPC User Guide, and the section called "Controlling Access to Kibana" (p. 99).

About Access Policies on VPC Domains

Placing your Amazon ES domain within a VPC provides an inherent, strong layer of security. When you create a domain with public access, the endpoint takes the following form:

```
https://search-domain-name-identifier.region.es.amazonaws.com
```

As the "public" label suggests, this endpoint is accessible from any internet-connected device, though you can (and should) control access to it (p. 30). If you access the endpoint in a web browser, you might receive a `Not Authorized` message, but the request reaches the domain.

When you create a domain with VPC access, the endpoint *looks* similar to a public endpoint:

```
https://vpc-domain-name-identifier.region.es.amazonaws.com
```

If you try to access the endpoint in a web browser, however, you might find that the request times out. To perform even basic `GET` requests, your computer must be able to connect to the VPC. This connection often takes the form of a VPN, managed network, or proxy server. For details on the various forms it can take, see Scenarios and Examples in the *Amazon VPC User Guide*. For a development-focused example, see the section called "Testing VPC Domains" (p. 120).

In addition to this connectivity requirement, VPCs let you manage access to the domain through security groups. For many use cases, this combination of security features is sufficient, and you might feel comfortable applying an open access policy to the domain.

Operating with an open access policy does *not* mean that anyone on the internet can access the Amazon ES domain. Rather, it means that if a request reaches the Amazon ES domain and the associated security groups permit it, the domain accepts the request without further security checks.

For an additional layer of security, we recommend using access policies that specify IAM users or roles. Applying these policies means that, for the domain to accept a request, the security groups must permit it *and* it must be signed with valid credentials.

> **Note**
> Because security groups already enforce IP-based access policies, you can't apply IP-based access policies to Amazon ES domains that reside within a VPC. If you use public access, IP-based policies are still available.

Testing VPC Domains

The enhanced security of a VPC can make connecting to your domain and running basic tests a real challenge. If you already have an Amazon ES VPC domain and would rather not create a VPN server, try the following process:

1. For your domain's access policy, choose **Do not require signing request with IAM credential**. You can always update this setting after you finish testing.
2. Create an Amazon Linux Amazon EC2 instance in the same VPC, subnet, and security group as your Amazon ES domain.

   Because this instance is for testing purposes and needs to do very little work, choose an inexpensive instance type like `t2.micro`. Assign the instance a public IP address and either create a new key pair or choose an existing one. If you create a new key, download it to your `~/.ssh` directory.

   To learn more about creating instances, see Getting Started with Amazon EC2 Linux Instances.
3. Add an internet gateway to your VPC.
4. In the route table for your VPC, add a new route. For **Destination**, specify a CIDR block that contains your computer's public IP address. For **Target**, specify the internet gateway you just created.

   For example, you might specify `123.123.123.123/32` for just your computer or `123.123.123.0/24` for a range of computers.
5. For the security group, specify two inbound rules:

Type	Protocol	Port Range	Source
SSH (22)	TCP (6)	22	*`your-cidr-block`*
HTTPS (443)	TCP (6)	443	*`your-security-group-id`*

   The first rule lets you SSH into your EC2 instance. The second allows the EC2 instance to communicate with the Amazon ES domain over HTTPS.

6. From the terminal, run the following command:

   ```
   ssh -i ~/.ssh/your-key.pem ec2-user@your-ec2-instance-public-ip -N -L 9200:vpc-your-amazon-es-domain.region.es.amazonaws.com:443
   ```

   This command creates an SSH tunnel that forwards requests to https://localhost:9200 to your Amazon ES domain through the EC2 instance. By default, Elasticsearch listens for traffic on port 9200. Specifying this port simulates a local Elasticsearch install, but use whichever port you'd like.

   The command provides no feedback and runs indefinitely. To stop it, press `Ctrl + C`.
7. Navigate to https://localhost:9200/_plugin/kibana/ in your web browser. You might need to acknowledge a security exception.

   Alternately, you can send requests to https://localhost:9200 using curl, Postman, or your favorite programming language.

   Tip
   If you encounter curl errors due to a certificate mismatch, try the `--insecure` flag.

Before You Begin: Prerequisites for VPC Access

Before you can enable a connection between a VPC and your new Amazon ES domain, you must do the following:

- **Create a VPC**

  To create your VPC, you can use the Amazon VPC console, the AWS CLI, or one of the AWS SDKs. You must create a subnet in the VPC, or two subnets if you enable zone awareness (p. 43). For more information, see Creating A VPC (p. 121). If you already have a VPC, you can skip this step.
- **Reserve IP addresses**

  Amazon ES enables the connection of a VPC to a domain by placing network interfaces in a subnet of the VPC. Each network interface is associated with an IP address. You must reserve a sufficient number of IP addresses in the subnet for the network interfaces. For more information, see Reserving IP Addresses in a VPC Subnet (p. 122).

Creating a VPC

To create your VPC, you can use one of the following: the Amazon VPC console, the AWS CLI, or one of the AWS SDKs. The VPC must have a subnet, or two subnets if you enable zone awareness (p. 43). The two subnets must be in different Availability Zones in the same region.

The following procedure shows how to use the Amazon VPC console to create a VPC with a public subnet, reserve IP addresses for the subnet, and create a security group to control access to your Amazon ES domain. For other VPC configurations, see Scenarios and Examples in the *Amazon VPC User Guide*.

To create a VPC (console)

1. Sign in to the AWS Management Console, and open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **VPC Dashboard**.
3. Choose **Start VPC Wizard**.
4. On the **Select a VPC Configuration** page, select **VPC with a Single Public Subnet**.

5. On the **VPC with a Single Public Subnet** page, keep the default options, and then choose **Create VPC**.
6. In the confirmation message that appears, choose **Close**.
7. If you intend to enable zone awareness (p. 43) for your Amazon ES domain, you must create a second subnet in a different Availability Zone in the same region. If you don't intend to enable zone awareness, skip to step 8.

   a. In the navigation pane, choose **Subnets.**
   b. Choose **Create Subnet**.
   c. In the **Create Subnet** dialog box, optionally create a name tag to help you identify the subnet later.
   d. For **VPC**, choose the VPC that you just created.
   e. For **Availability Zone**, choose an Availability Zone that differs from that of the first subnet. The Availability Zones for both subnets must be in the same region.
   f. For **IPv4 CIDR block**, configure a CIDR block large enough to provide sufficient IP addresses for Amazon ES to use during maintenance activities. For more information, see Reserving IP Addresses in a VPC Subnet (p. 122).

      Note
      Amazon ES domains using VPC access don't support IPv6 addresses. You can use a VPC that has IPv6 enabled, but the ENIs will have IPv4 addresses.
   g. Choose **Yes, Create**.
8. In the navigation pane, choose **Subnets**.
9. In the list of subnets, find your subnet (or subnets, if you created a second subnet in step 7). In the **Available IPv4** column, confirm that you have a sufficient number of IPv4 addresses.
10. Make a note of the subnet ID and Availability Zone. You need this information later when you launch your Amazon ES domain and add an Amazon EC2 instance to your VPC.
11. Create an Amazon VPC security group. You use this security group to control access to your Amazon ES domain.

    a. In the navigation pane, choose **Security Groups**.
    b. Choose **Create Security Group**.
    c. In the **Create Security Group** dialog box, type a name tag, a group name, and a description. For **VPC**, choose the ID of your VPC.
    d. Choose **Yes, Create**.
12. Define a network ingress rule for your security group. This rule allows you to connect to your Amazon ES domain.

    a. In the navigation pane, choose **Security Groups**, and then select the security group that you just created.
    b. At the bottom of the page, choose the **Inbound Rules** tab.
    c. Choose **Edit**, and then choose **HTTPS (443)**.
    d. Choose **Save**.

Now you are ready to launch an Amazon ES domain (p. 10) in your Amazon VPC.

Reserving IP Addresses in a VPC Subnet

Amazon ES connects a domain to a VPC by placing network interfaces in a subnet of the VPC (or two subnets of the VPC if you enable zone awareness (p. 43)). Each network interface is associated with an IP address. Before you create your Amazon ES domain, you must have a sufficient number of IP addresses available in the VPC subnet to accommodate the network interfaces.

The number of IP addresses that Amazon ES requires depends on the following:

- Number of data nodes in your domain. (Master nodes are not included in the number.)
- Whether you enable zone awareness (p. 43). If you enable zone awareness, you need only half the number of IP addresses per subnet that you need if you don't enable zone awareness.

Here is the basic formula: The number of IP addresses reserved in each subnet is three times the number of nodes, divided by two if zone awareness is enabled.

Examples

- If a domain has 10 data nodes and zone awareness is enabled, the IP count is 10 / 2 * 3 = 15.
- If a domain has 10 data nodes and zone awareness is disabled, the IP count is 10 * 3 = 30.

When you create the domain, Amazon ES reserves the IP addresses. You can see the network interfaces and their associated IP addresses in the **Network Interfaces** section of the Amazon EC2 console at https://console.aws.amazon.com/ec2/. The **Description** column shows which Amazon ES domain the network interface is associated with.

> **Tip**
> We recommend that you create dedicated subnets for the Amazon ES reserved IP addresses. By using dedicated subnets, you avoid overlap with other applications and services and ensure that you can reserve additional IP addresses if you need to scale your cluster in the future. To learn more, see Creating a Subnet in Your VPC.

Service-Linked Role for VPC Access

A service-linked role is a unique type of IAM role that delegates permissions to a service so that it can create and manage resources on your behalf. Amazon ES requires a service-linked role to access your VPC, create the domain endpoint, and place network interfaces in a subnet of your VPC.

Amazon ES automatically creates the role when you use the Amazon ES console to create a domain within a VPC. For this automatic creation to succeed, you must have permissions for the `iam:CreateServiceLinkedRole` action. To learn more, see Service-Linked Role Permissions in the *IAM User Guide*.

After Amazon ES creates the role, you can view it (`AWSServiceRoleForAmazonElasticsearchService`) using the IAM console.

> **Note**
> If you create a domain that uses a public endpoint, Amazon ES doesn’t need the service-linked role and doesn't create it.

For full information on this role's permissions and how to delete it, see *Using Service-Linked Roles* (p. 256).

Migrating from Public Access to VPC Access

When you create a domain, you specify whether it should have a public endpoint or reside within a VPC. Once created, you cannot switch from one to the other. Instead, you must create a new domain and either manually reindex or migrate your data. Snapshots offer a convenient means of migrating data. For information about taking and restoring snapshots, see *Working with Index Snapshots* (p. 87).

Amazon VPC Documentation

Amazon VPC has its own set of documentation to describe how to create and use your Amazon VPC. The following table provides links to the Amazon VPC guides.

Description	Documentation
How to get started using Amazon VPC	Amazon VPC Getting Started Guide
How to use Amazon VPC through the AWS Management Console	Amazon VPC User Guide
Complete descriptions of all the Amazon VPC commands	Amazon EC2 Command Line Reference (The Amazon VPC commands are part of the Amazon EC2 reference.)
Complete descriptions of the Amazon VPC API actions, data types, and errors	Amazon EC2 API Reference (The Amazon VPC API actions are part of the Amazon EC2 reference.)
Information for the network administrator who configures the gateway at your end of an optional IPsec VPN connection	Amazon VPC Network Administrator Guide

For more detailed information about Amazon Virtual Private Cloud, see Amazon Virtual Private Cloud.

Amazon Elasticsearch Service Best Practices

These topics address some considerations for operating Amazon Elasticsearch Service domains and provide general guidelines that apply to many use cases.

Topics

Sizing Amazon ES Domains

No surefire method of sizing Amazon ES domains exists, but by starting with an understanding of your storage needs, the service, and Elasticsearch itself, you can make an educated initial estimate on your hardware needs. This estimate can serve as a useful starting point for the most critical aspect of sizing domains: testing them with representative workloads and monitoring their performance.

Topics

Calculating Storage Requirements

Most Elasticsearch workloads fall into one of two broad categories:

- Long-lived index: You write code that processes data into one or more Elasticsearch indices and then updates those indices periodically as the source data changes. Some common examples are website, document, and e-commerce search.
- Rolling indices: Data continuously flows into a set of temporary indices, with an indexing period and retention window, such as a set of daily indices that is retained for two weeks. Some common examples are log analytics, time-series processing, and clickstream analytics.

For long-lived index workloads, you can examine the source data on disk and easily determine how much storage space it consumes. If the data comes from multiple sources, just add those sources together.

For rolling indices, you can multiply the amount of data generated during a representative time period by the retention period. For example, if you generate 200 MB of log data per hour, that's 4.8 GB per day, which is 67 GB of data at any given time if you have a two-week retention period.

The size of your source data, however, is just one aspect of your storage requirements. You also have to consider the following:

1. Number of replicas: Each replica is a full copy of an index and needs the same amount of disk space. By default, each Elasticsearch index has one replica. We recommend at least one to prevent data loss. Replicas also improve search performance, so you might want more if you have a read-heavy workload.

2. Elasticsearch indexing overhead: The on-disk size of an index varies, but is often 10% larger than the source data. After indexing your data, you can use the `_cat/indices` API and `pri.store.size` value to calculate the exact overhead. The `_cat/allocation` API also provides a useful summary.
3. Operating system reserved space: By default, Linux reserves 5% of the file system for the `root` user for critical processes, system recovery, and to safeguard against disk fragmentation problems.
4. Amazon ES overhead: Amazon ES reserves 20% of the storage space of each instance (up to 20 GB) for segment merges, logs, and other internal operations.

   Because of this 20 GB maximum, the total amount of reserved space can vary dramatically depending on the number of instances in your domain. For example, a domain might have three `m4.xlarge.elasticsearch` instances, each with 500 GB of storage space, for a total of 1.5 TB. In this case, the total reserved space is only 60 GB. Another domain might have 10 `m3.medium.elasticsearch` instances, each with 100 GB of storage space, for a total of 1 TB. Here, the total reserved space is 200 GB, even though the first domain is 50% larger.

   In the following formula, we apply a "worst-case" estimate for overhead that is accurate for domains with less than 100 GB of storage space per instance and over-allocates for larger instances.

In summary, if you have 67 GB of data at any given time and want one replica, your *minimum* storage requirement is closer to 67 * 2 * 1.1 / 0.95 / 0.8 = 194 GB. You can generalize this calculation as follows:

Source Data * (1 + Number of Replicas) * (1 + Indexing Overhead) / (1 - Linux Reserved Space) / (1 - Amazon ES Overhead) = Minimum Storage Requirement

Or you can use this simplified version:

Source Data * (1 + Number of Replicas) * 1.45 = Minimum Storage Requirement

Insufficient storage space is one of the most common causes of cluster instability, so you should cross-check the numbers when you choose instance types, instance counts, and storage volumes (p. 127).

> **Note**
> If your minimum storage requirement exceeds 1 PB, see *Petabyte Scale* (p. 132).

Choosing the Number of Shards

After you understand your storage requirements, you can investigate your indexing strategy. Each Elasticsearch index is split into some number of shards. Because you can't easily change the number of primary shards for an existing index, you should decide about shard count *before* indexing your first document.

The overarching goal of choosing a number of shards is to distribute an index evenly across all data nodes in the cluster. However, these shards shouldn't be too large or too numerous. A good rule of thumb is to try to keep shard size between 10–50 GB. Large shards can make it difficult for Elasticsearch to recover from failure, but because each shard uses some amount of CPU and memory, having too many small shards can cause performance issues and out of memory errors. In other words, shards should be small enough that the underlying Amazon ES instance can handle them, but not so small that they place needless strain on the hardware.

For example, suppose you have 67 GB of data. You don't expect that number to increase over time, and you want to keep your shards around 30 GB each. Your number of shards therefore should be approximately 67 * 1.1 / 30 = 3. You can generalize this calculation as follows:

(Source Data + Room to Grow) * (1 + Indexing Overhead) / Desired Shard Size = Approximate Number of Primary Shards

This equation helps compensate for growth over time. If you expect those same 67 GB of data to quadruple over the next year, the approximate number of shards is (67 + 201) * 1.1 / 30 = 10. Remember, though, you don't have those extra 201 GB of data *yet*. Check to make sure this preparation for the future doesn't create unnecessarily tiny shards that consume huge amounts of CPU and memory in the present. In this case, 67 * 1.1 / 10 shards = 7.4 GB per shard, which will consume extra resources and is below the recommended size range. You might consider the more middle-of-the-road approach of six shards, which leaves you with 12 GB shards today and 49 GB shards in the future. Then again, you might prefer to start with three 30 GB shards and reindex your data when the shards exceed 50 GB.

> **Note**
> By default, Elasticsearch indices are split into five primary shards. You can specify different settings when you create an index (p. 59).

Choosing Instance Types and Testing

After you calculate your storage requirements and choose the number of shards that you need, you can start to make hardware decisions. Hardware requirements vary dramatically by workload, but we can still offer some basic recommendations.

In general, the storage limits (p. 225) for each instance type map to the amount of CPU and memory you might need for light workloads. For example, an `m4.large.elasticsearch` instance has a maximum EBS volume size of 512 GB, 2 vCPU cores, and 8 GB of memory. If your cluster has many shards, performs taxing aggregations, updates documents frequently, or processes a large number of queries, those resources might be insufficient for your needs. If you believe your cluster falls into one of these categories, try starting with a configuration closer to 2 vCPU cores and 8 GB of memory for every 100 GB of your storage requirement.

> **Tip**
> For a summary of the hardware resources that are allocated to each instance type, see Amazon Elasticsearch Service Pricing.

Still, even those resources might be insufficient. Some Elasticsearch users report that they need many times those resources to fulfill their requirements. Finding the right hardware for your workload means making an educated initial estimate, testing with representative workloads, adjusting, and testing again:

1. To start, we recommend a minimum of three instances to avoid potential Elasticsearch issues, such as the split brain issue. If you have three dedicated master nodes (p. 128), we still recommend a minimum of two data nodes for replication.
2. If you have a 184 GB storage requirement and the recommended minimum number of three instances, you use the equation 184 / 3 = 61 GB to find the amount of storage that each instance needs. In this example, you might select three `m4.large.elasticsearch` instances for your cluster, each using a 90 GB EBS storage volume so that you have a safety net and some room for growth over time. This configuration provides 6 vCPU cores and 24 GB of memory, so it's suited to lighter workloads.

   For a more substantial example, consider a 14 TB storage requirement and a heavy workload. In this case, you might choose to begin testing with 2 * 140 = 280 vCPU cores and 8 * 140 = 1120 GB of memory. These numbers work out to approximately 18 `i3.4xlarge.elasticsearch` instances. If you don't need the fast, local storage, you could also test 18 `r4.4xlarge.elasticsearch` instances, each using a 900 GB EBS storage volume.

   If your cluster includes hundreds of terabytes of data, see *Petabyte Scale* (p. 132).
3. After configuring the cluster, you can add your index (p. 59), perform some representative client testing using a realistic dataset, and monitor CloudWatch metrics (p. 46) to see how the cluster handles the workload.
4. If performance satisfies your needs, tests succeed, and CloudWatch metrics are normal, the cluster is ready to use. Remember to set CloudWatch alarms (p. 130) to detect unhealthy resource usage.

   If performance isn't acceptable, tests fail, or `CPUUtilization` or `JVMMemoryPressure` are high, you might need to choose a different instance type (or add instances) and continue testing. As you add instances, Elasticsearch automatically rebalances the distribution of shards throughout the cluster.

Because it is easier to measure the excess capacity in an overpowered cluster than the deficit in an underpowered one, we recommend starting with a larger cluster than you think you need, testing, and scaling down to an efficient cluster that has the extra resources to ensure stable operations during periods of increased activity.

Production clusters or clusters with complex states benefit from dedicated master nodes (p. 128), which improve performance and cluster reliability.

Dedicated Master Nodes

Amazon Elasticsearch Service uses *dedicated master nodes* to increase cluster stability. A dedicated master node performs cluster management tasks, but does not hold data or respond to data upload requests. This offloading of cluster management tasks increases the stability of your domain.

We recommend that you allocate **three** dedicated master nodes for each production Amazon ES domain:

1. One dedicated master node means that you have no backup in the event of a failure.
2. Two dedicated master nodes means that your cluster does not have the necessary quorum of nodes to elect a new master node in the event of a failure.

   A quorum is Number of Dedicated Master Nodes / 2 + 1 (rounded down to the nearest whole number), which Amazon ES sets to `discovery.zen.minimum_master_nodes` when you create your domain.

   In this case, 2 / 2 + 1 = 2. Because one dedicated master node has failed and only one backup exists, the cluster does not have a quorum and cannot elect a new master.
3. Three dedicated master nodes, the recommended number, provides two backup nodes in the event of a master node failure and the necessary quorum (2) to elect a new master.
4. Four dedicated master nodes is no better than three and can cause issues if you use zone awareness (p. 43).
   - If one master node fails, you have the quorum (3) to elect a new master. If two nodes fail, you lose that quorum, just as you do with three dedicated master nodes.
   - If each Availability Zone has two dedicated master nodes and the zones are unable to communicate with each other, neither zone has the quorum to elect a new master.
5. Having five dedicated master nodes works as well as three and allows you to lose two nodes while maintaining a quorum, but because only one dedicated master node is active at any given time, this configuration means paying for four idle nodes. Many customers find this level of failover protection excessive.

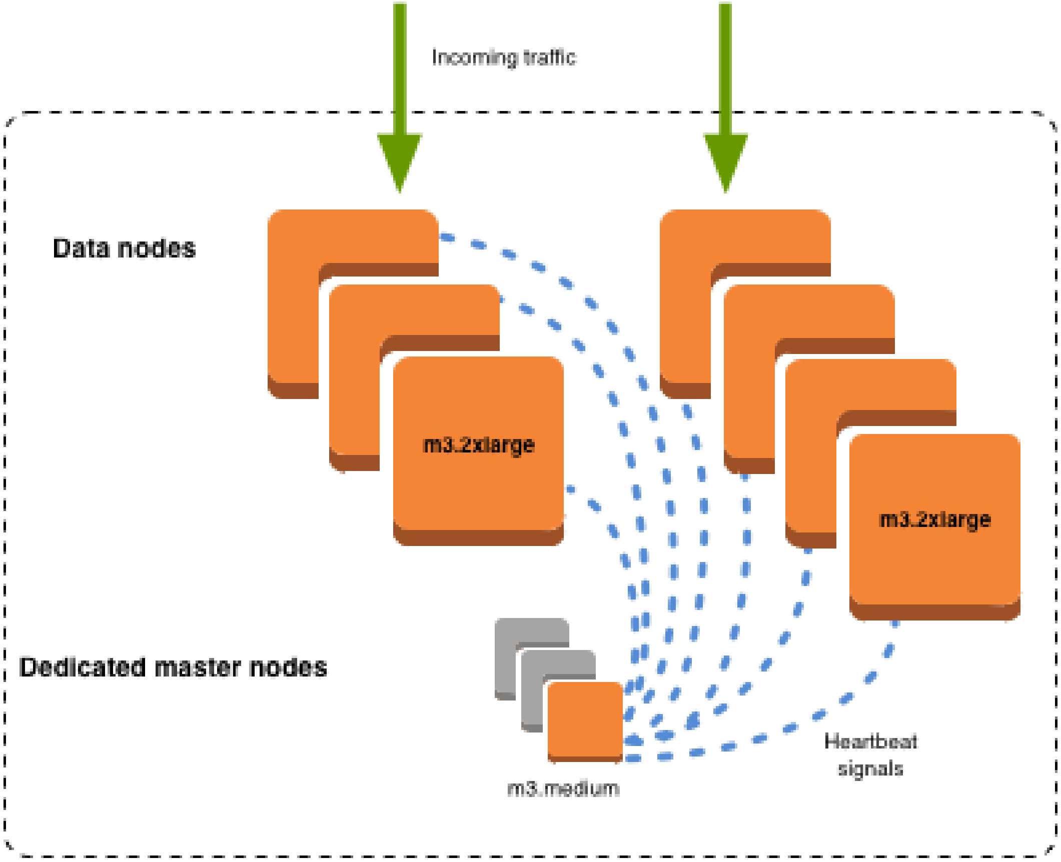
> **Note**
> If your cluster does not have the necessary quorum to elect a new master node, write *and* read requests to the cluster both fail. This behavior differs from the Elasticsearch default.

Dedicated master nodes perform the following cluster management tasks:

- Track all nodes in the cluster
- Track the number of indices in the cluster
- Track the number of shards belonging to each index
- Maintain routing information for nodes in the cluster
- Update the cluster state after state changes, such as creating an index and adding or removing nodes in the cluster

- Replicate changes to the cluster state across all nodes in the cluster
- Monitor the health of all cluster nodes by sending *heartbeat signals*, periodic signals that monitor the availability of the data nodes in the cluster

The following illustration shows an Amazon ES domain with ten instances. Seven of the instances are data nodes and three are dedicated master nodes. Only one of the dedicated master nodes is active; the two gray dedicated master nodes wait as backup in case the active dedicated master node fails. All data upload requests are served by the seven data nodes, and all cluster management tasks are offloaded to the active dedicated master node.



Although dedicated master nodes do not process search and query requests, their size is highly correlated with the number of instances, indices, and shards that they can manage. For production clusters, we recommend the following instance types for dedicated master nodes. These recommendations are based on typical workloads and can vary based on your needs.

Instance Count	Recommended Minimum Dedicated Master Instance Type
5–10	`m3.medium.elasticsearch`
10–20	`m4.large.elasticsearch`
20–50	`c4.xlarge.elasticsearch`

Instance Count	Recommended Minimum Dedicated Master Instance Type
50–100	`c4.2xlarge.elasticsearch`

- For recommendations on dedicated master nodes for large clusters, see *Petabyte Scale* (p. 132).
- For information about how certain configuration changes can affect dedicated master nodes, see the section called “About Configuration Changes” (p. 42).
- For clarification on instance count limits, see the section called “Cluster and Instance Limits” (p. 225).
- For more information about specific instance types, including vCPU, memory, and pricing, see Amazon Elasticsearch Instance Prices.

Recommended CloudWatch Alarms

CloudWatch alarms perform an action when a CloudWatch metric exceeds a specified value for some amount of time. For example, you might want AWS to email you if your cluster health status is `red` for longer than one minute. This section includes some recommended alarms and how to respond to them.

For more information about setting alarms, see Creating Amazon CloudWatch Alarms in the *Amazon CloudWatch User Guide*.

Alarm	Issue
`ClusterStatus.red` maximum is >= 1 for 1 minute, 1 consecutive time	At least one primary shard and its replicas are not allocated to a node. See the section called “Red Cluster Status” (p. 143).
`ClusterStatus.yellow` maximum is >= 1 for 1 minute, 1 consecutive time	At least one replica shard is not allocated to a node. See the section called “Yellow Cluster Status” (p. 145).
`FreeStorageSpace` minimum is <= 20000 for 1 minute, 1 consecutive time	A node in your cluster is down to 20 GB of free storage space. See the section called “Lack of Available Storage Space” (p. 145). This value is in MB, so rather than 20000, we recommend setting it to 25% of the storage space for each node.
`ClusterIndexWritesBlocked` is >= 1 for 5 minutes, 1 consecutive time	Your cluster is blocking write requests. See the section called “ClusterBlockException” (p. 145).
`Nodes` minimum is < *x* for 1 day, 1 consecutive time	*x* is the number of nodes in your cluster. This alarm indicates that at least one node in your cluster has been unreachable for one day. See the section called “Failed Cluster Nodes” (p. 146).
`AutomatedSnapshotFailure` maximum is >= 1 for 1 minute, 1 consecutive time	An automated snapshot failed. This failure is often the result of a red cluster health status. See the section called “Red Cluster Status” (p. 143). For a summary of all automated snapshots and some information about failures, you can also try the following: `GET domain_endpoint/_snapshot/cs-automated/_all`

Alarm	Issue
`CPUUtilization` average is >= 80% for 15 minutes, 3 consecutive times	100% CPU utilization isn't uncommon, but *sustained* high averages are problematic. Consider using larger instance types or adding instances.
`JVMMemoryPressure` maximum is >= 80% for 15 minutes, 1 consecutive time	The cluster could encounter out of memory errors if usage increases. Consider scaling vertically. Amazon ES uses half of an instance's RAM for the Java heap, up to a heap size of 32 GB. You can scale instances vertically up to 64 GB of RAM, at which point you can scale horizontally by adding instances.
`MasterCPUUtilization` average is >= 50% for 15 minutes, 3 consecutive times	Consider using larger instance types for your dedicated master nodes (p. 128). Because of their role in cluster stability and blue/green deployments (p. 42), dedicated master nodes should have lower average CPU usage than data nodes.
`MasterJVMMemoryPressure` maximum is >= 80% for 15 minutes, 1 consecutive time	
`KMSKeyError` is >= 1 for 1 minute, 1 consecutive time	The KMS encryption key that is used to encrypt data at rest in your domain is disabled. Re-enable it to restore normal operations. To learn more, see the section called "Encryption at Rest" (p. 134).
`KMSKeyInaccessible` is >= 1 for 1 minute, 1 consecutive time	The KMS encryption key that is used to encrypt data at rest in your domain has been deleted or has revoked its grants to Amazon ES. You can't recover domains that are in this state, but if you have a manual snapshot, you can use it to migrate to a new domain. To learn more, see the section called "Encryption at Rest" (p. 134).

Note
If you just want to *view* metrics, see Monitoring CloudWatch Metrics (p. 46).

Petabyte Scale for Amazon Elasticsearch Service

Amazon Elasticsearch Service offers domain storage of up to 1.5 PB. You can configure a domain with 100 `i3.16xlarge.elasticsearch` instance types, each with 15 TB of storage. Because of the sheer difference in scale, recommendations for domains of this size differ from our general recommendations (p. 125). This section discusses considerations for creating domains, costs, storage, shard size, and dedicated master nodes. Despite frequent references to the `i3` instance types, the shard size and dedicated master node recommendations in this section apply to any domain approaching petabyte scale.

Creating domains

Domains of this size exceed the default limit of 20 instances per domain. To request a service limit increase of up to 100 instances per domain, open a case at the AWS Support Center.

Pricing

Before creating a domain of this size, check the Amazon Elasticsearch Service Pricing page to ensure that the associated costs match your expectations.

Storage

The `i3` instance types are specifically designed to provide fast, local non-volatile memory express (NVMe) storage. Because this local storage tends to offer considerable performance benefits when compared to Amazon Elastic Block Store, EBS volumes are not an option when you select these instance types in Amazon ES.

Shard size and count

A common Elasticsearch guideline is not to exceed 50 GB per shard. Given the number of shards necessary to accommodate large domains and the resources available to `i3.16xlarge.elasticsearch` instances, we recommend a shard size of 100 GB.

For example, if you have 450 TB of source data and want one replica, your *minimum* storage requirement is closer to 450 TB * 2 * 1.1 / 0.95 = 1.04 PB. For an explanation of this calculation, see the section called “Calculating Storage Requirements” (p. 125). Although 1.04 PB / 15 TB = 70 instances, you might select 80 or more `i3.16xlarge.elasticsearch` instances to give yourself a storage safety net and account for some variance in the amount of data over time. Each instance adds another 20 GB to your minimum storage requirement, but for disks of this size, those 20 GB are almost negligible.

Controlling the number of shards is tricky. Elasticsearch users often rotate indices on a daily basis and retain data for a week or two. In this situation, you might find it useful to distinguish between "active" and "inactive" shards. Active shards are, well, actively being written to or read from. Inactive shards might service the occasional read request, but are largely idle. In general, you should keep the number of active shards below a few thousand. As the number of active shards approaches 10,000, considerable performance and stability risks emerge.

To calculate the number of primary shards, use this formula: 450,000 GB * 1.1 / 100 GB per shard = 4,950 shards. Doubling that number to account for replicas is 9,900 shards, which represents a major concern if all shards are active. But if you rotate indices and only 1/7th or 1/14th of the shards are active on any given day (1414 or 707 shards, respectively), the cluster might work well. As always, the most important step of sizing and configuring your domain is to perform representative client testing using a realistic data set.

Dedicated master nodes

We recommend that you allocate three dedicated master nodes to each production Amazon ES domain. Rather than our usual guidelines for dedicated master nodes (p. 128), however, we recommend more powerful instance types for domains of this size. The following table shows recommended instance types for dedicated master nodes for large domains.

Instance Count	Recommended Minimum Dedicated Master Instance Type
5–10	`m4.large.elasticsearch`
10–20	`c4.xlarge.elasticsearch`
20–50	`c4.2xlarge.elasticsearch`
50–100	`c4.4xlarge.elasticsearch`

Encryption in Amazon Elasticsearch Service

These topics cover how to enable encryption of data at rest and node-to-node encryption for Amazon Elasticsearch Service domains.

Topics

Encryption of Data at Rest for Amazon Elasticsearch Service

Amazon ES domains offer encryption of data at rest, a security feature that helps prevent unauthorized access to your data. The feature uses AWS Key Management Service (AWS KMS) to store and manage your encryption keys. If enabled, it encrypts the following aspects of a domain:

- Indices
- Automated snapshots
- Elasticsearch logs
- Swap files
- All other data in the application directory

The following are *not* encrypted when you enable encryption of data at rest, but you can take additional steps to protect them:

- Manual snapshots: Currently, you can't use KMS master keys to encrypt manual snapshots. You can, however, use server-side encryption with S3-managed keys to encrypt the bucket that you use as a snapshot repository. For instructions, see the section called “Registering a Manual Snapshot Repository” (p. 89).
- Slow logs: If you publish slow logs (p. 25) and want to encrypt them, you can encrypt their CloudWatch Logs log group using the same AWS KMS master key as the Amazon ES domain. For more information, see Encrypt Log Data in CloudWatch Logs Using AWS KMS in the Amazon CloudWatch Logs User Guide.

To learn how to create AWS KMS master keys, see Creating Keys in the *AWS Key Management Service Developer Guide*.

Enabling Encryption of Data at Rest

By default, domains don't encrypt data at rest, and you can't configure existing domains to use the feature. To enable the feature, you must create another domain (p. 10) and migrate your data. Encryption of data at rest requires Elasticsearch 5.1 or later.

To use the Amazon ES console to create a domain that encrypts data at rest, you must have read-only permissions to AWS KMS, such as the following identity-based policy:

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

If you want to use a key other than **(Default) aws/es**, you must also have permissions to create grants for the key. These permissions typically take the form of a resource-based policy that you specify when you create the key.

If you want to keep your key exclusive to Amazon ES, you can add the `kms:ViaService` condition to the key policy:

```
"Condition": {
  "StringEquals": {
    "kms:ViaService": "es.us-west-1.amazonaws.com"
  },
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
```

For more information, see Using Key Policies in AWS KMS in the *AWS Key Management Service Developer Guide*.

Warning
If you delete the key that you used to encrypt a domain, the domain becomes inaccessible. The Amazon ES team can't help you recover your data. AWS KMS deletes master keys only after a waiting period of at least seven days, so the Amazon ES team might contact you if they detect that your domain is at risk.

Disabling Encryption of Data at Rest

After you configure a domain to encrypt data at rest, you can't disable the setting. Instead, you can take a manual snapshot (p. 87) of the existing domain, create another domain (p. 10), migrate your data, and delete the old domain.

Monitoring Domains That Encrypt Data at Rest

Domains that encrypt data at rest have two additional metrics: `KMSKeyError` and `KMSKeyInaccessible`. For full descriptions of these metrics, see the section called “Cluster Metrics” (p. 46). You can view them using either the Amazon ES console or the Amazon CloudWatch console.

Tip
Each metric represents a significant problem for a domain, so we recommend that you create CloudWatch alarms for both. For more information, see the section called “Recommended CloudWatch Alarms” (p. 130).

Other Considerations

- Automatic key rotation preserves the properties of your AWS KMS master keys, so the rotation has no effect on your ability to access your Elasticsearch data. Encrypted Amazon ES domains do not support manual key rotation, which involves creating a new master key and updating any references to the old key. To learn more, see Rotating Customer Master Keys in the *AWS Key Management Service Developer Guide*.
- Certain instance types do not support encryption of data at rest. For details, see the section called "Supported Instance Types" (p. 150).
- Encryption of data at rest is not available in the cn-north-1 (Beijing) and cn-northwest-1 (Ningxia) Region.
- Kibana still works on domains that encrypt data at rest.
- Domains that encrypt data at rest use a different repository name for their automated snapshots. For more information, see the section called "Restoring Snapshots" (p. 92).
- Encrypting an Amazon ES domain requires two grants, and each encryption key has a limit of 500 grants per principal. This limit means that the maximum number of Amazon ES domains that you can encrypt using a single key is 250. Currently, Amazon ES supports a maximum of 100 domains per account, so this grant limit is of no consequence. If the domain limit per account increases, however, the grant limit might become relevant.

  If you need to encrypt more than 250 domains at that time, you can create additional keys. Keys are regional, not global, so if you operate in more than one AWS Region, you already need multiple keys.

Node-to-node Encryption for Amazon Elasticsearch Service

Node-to-node encryption provides an additional layer of security on top of the default features of Amazon ES.

Each Amazon ES domain—regardless of whether the domain uses VPC access—resides within its own, dedicated VPC. This architecture prevents potential attackers from intercepting traffic between Elasticsearch nodes and keeps the cluster secure. By default, however, traffic within the VPC is unencrypted. Node-to-node encryption enables TLS encryption for all communications within the VPC.

If you send data to Amazon ES over HTTPS, node-to-node encryption helps ensure that your data remains encrypted as Elasticsearch distributes (and redistributes) it throughout the cluster. If data arrives unencrypted over HTTP, Amazon ES encrypts it after it reaches the cluster.

Enabling Node-to-node Encryption

By default, domains do not use node-to-node encryption, and you can't configure existing domains to use the feature. To enable the feature, you must create another domain (p. 10) and migrate your data (p. 96). Node-to-node encryption requires Elasticsearch 6.0 or later.

Disabling Node-to-node Encryption

After you configure a domain to use node-to-node encryption, you can't disable the setting. Instead, you can take a manual snapshot (p. 87) of the encrypted domain, create another domain (p. 10), migrate your data, and delete the old domain.

Other Considerations

- Node-to-node encryption is not available in the cn-north-1 (Beijing) and cn-northwest-1 (Ningxia) Region.
- Kibana still works on domains that use node-to-node encryption.

Using Curator to Rotate Data in Amazon Elasticsearch Service

This chapter has sample code for using AWS Lambda and Curator to manage indices and snapshots. Curator offers numerous filters to help you identity indices and snapshots that meet certain criteria, such as indices created more than 60 days ago or snapshots that failed to complete.

Although Curator is often used as a command line interface (CLI), it also features a Python API, which means that you can use it within Lambda functions.

For information about configuring Lambda functions and creating deployment packages, see the section called "Loading Streaming Data into Amazon ES from Amazon S3" (p. 70). For even more information, see the AWS Lambda Developer Guide. This chapter contains only sample code, basic settings, triggers, and permissions.

Topics

Sample Code

The following sample code uses Curator and the official Python Elasticsearch client to delete any index whose name contains a time stamp indicating that the data is more than 30 days old. For example, if an index name is `my-logs-2014.03.02`, the index is deleted. Deletion occurs even if you create the index today, because this filter uses the name of the index to determine its age.

The code also contains some commented-out examples of other common filters, including one that determines age by creation date. The AWS SDK for Python (Boto 3) and requests-aws4auth library sign the requests to Amazon ES.

> **Warning**
> Both code samples in this section delete data—potentially a lot of data. Modify and test each sample on a non-critical domain until you're satisfied with its behavior.

Index Deletion

```
import boto3
from requests_aws4auth import AWS4Auth
from elasticsearch import Elasticsearch, RequestsHttpConnection
import curator

host = '' # For example, search-my-domain.region.es.amazonaws.com
region = '' # For example, us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Lambda execution starts here.
def lambda_handler(event, context):
    # Build the Elasticsearch client.
    es = Elasticsearch(
        hosts = [{'host': host, 'port': 443}],
        http_auth = awsauth,
        use_ssl = True,
        verify_certs = True,
        connection_class = RequestsHttpConnection
    )

    # A test document.
    document = {
        "title": "Moneyball",
        "director": "Bennett Miller",
        "year": "2011"
    }

    # Index the test document so that we have an index that matches the timestring pattern.
    # You can delete this line and the test document if you already created some test
 indices.
    es.index(index="movies-2017.01.31", doc_type="movie", id="1", body=document)

    index_list = curator.IndexList(es)

    # Filters by age, anything with a time stamp older than 30 days in the index name.
    index_list.filter_by_age(source='name', direction='older', timestring='%Y.%m.%d',
 unit='days', unit_count=30)

    # Filters by naming prefix.
    # index_list.filter_by_regex(kind='prefix', value='my-logs-2017')

    # Filters by age, anything created more than one month ago.
    # index_list.filter_by_age(source='creation_date', direction='older', unit='months',
 unit_count=1)

    print("Found %s indices to delete" % len(index_list.indices))

    # If our filtered list contains any indices, delete them.
    if index_list.indices:
        curator.DeleteIndices(index_list).do_action()
```

You must update the values for `host` and `region`.

The next code sample deletes any snapshot that is more than two weeks old. It also takes a new snapshot.

Snapshot Deletion

```
import boto3
from datetime import datetime
from requests_aws4auth import AWS4Auth
from elasticsearch import Elasticsearch, RequestsHttpConnection
import logging
import curator

# Adding a logger isn't strictly required, but helps with understanding Curator's requests
 and debugging.
logger = logging.getLogger('curator')
logger.addHandler(logging.StreamHandler())
logger.setLevel(logging.INFO)

host = '' # For example, search-my-domain.region.es.amazonaws.com
region = '' # For example, us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
```

```
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

now = datetime.now()
# Clunky, but this approach keeps colons out of the URL.
date_string = '-'.join((str(now.year), str(now.month), str(now.day), str(now.hour),
 str(now.second)))

snapshot_name = 'my-snapshot-prefix-' + date_string
repository_name = 'my-repo'

# Lambda execution starts here.
def lambda_handler(event, context):

    # Build the Elasticsearch client.
    es = Elasticsearch(
        hosts = [{'host': host, 'port': 443}],
        http_auth = awsauth,
        use_ssl = True,
        verify_certs = True,
        connection_class = RequestsHttpConnection,
        timeout = 120 # Deleting snapshots can take a while, so keep the connection open
 for long enough to get a response.
    )

    try:
        # Get all snapshots in the repository.
        snapshot_list = curator.SnapshotList(es, repository=repository_name)

        # Filter by age, any snapshot older than two weeks.
        # snapshot_list.filter_by_age(source='creation_date', direction='older',
 unit='weeks', unit_count=2)

        # Delete the old snapshots.
        curator.DeleteSnapshots(snapshot_list, retry_interval=30,
 retry_count=3).do_action()
    except (curator.exceptions.SnapshotInProgress, curator.exceptions.NoSnapshots,
 curator.exceptions.FailedExecution) as e:
        print(e)

    # Split into two try blocks. We still want to try and take a snapshot if deletion
 failed.
    try:
        # Get the list of indices.
        # You can filter this list if you didn't want to snapshot all indices.
        index_list = curator.IndexList(es)

        # Take a new snapshot. This operation can take a while, so we don't want to wait
 for it to complete.
        curator.Snapshot(index_list, repository=repository_name, name=snapshot_name,
 wait_for_completion=False).do_action()
    except (curator.exceptions.SnapshotInProgress, curator.exceptions.FailedExecution) as
 e:
        print(e)
```

You must update the values for `host`, `region`, `snapshot_name`, and `repository_name`. If the output is too verbose for your taste, you can change `logging.INFO` to `logging.WARN`.

Because taking and deleting snapshots can take a while, this code is more sensitive to connection and Lambda timeouts—hence the extra logging code. In the Elasticsearch client, you can see that we set the timeout to 120 seconds. If the `DeleteSnapshots` function takes longer to get a response from the Amazon ES domain, you might need to increase this value. You must also increase the Lambda function timeout from its default value of three seconds. For a recommended value, see the section called "Basic Settings" (p. 141).

Basic Settings

We recommend the following settings for the code samples in this chapter.

Sample Code	Memory	Timeout
Index Deletion	128 MB	10 seconds
Snapshot Deletion	128 MB	3 minutes

Triggers

Rather than reacting to some event (such as a file upload to Amazon S3), these functions are meant to be scheduled. You might prefer to run these functions more or less frequently.

Sample Code	Service	Rule Type	Example Expression
Index Deletion	CloudWatch Events	Schedule expression	rate(1 day)
Snapshot Deletion	CloudWatch Events	Schedule expression	rate(4 hours)

Permissions

Both Lambda functions in this chapter need the basic logging permissions that all Lambda functions need, plus HTTP method permissions for the Amazon ES domain:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:CreateLogGroup",
      "Resource": "arn:aws:logs:us-west-1:123456789012:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-west-1:123456789012:log-group:/aws/lambda/your-lambda-function:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpPost",
        "es:ESHttpGet",
        "es:ESHttpPut",
        "es:ESHttpDelete"
      ],
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/my-domain/*"
    }
  ]
}
```

Amazon Elasticsearch Service Troubleshooting

This section describes how to identity and solve common Amazon Elasticsearch Service issues. Consult the information in this section before contacting AWS Support.

Can't Access Kibana

The Kibana endpoint doesn't support signed requests. If the access control policy for your domain only grants access to certain IAM users or roles and you haven't configured *Authentication for Kibana* (p. 104), you might receive the following error when you attempt to access Kibana:

```
"User: anonymous is not authorized to perform: es:ESHttpGet"
```

If your Amazon ES domain uses VPC access, you might not receive this error. Instead, the request might time out. To learn more about correcting this issue and the various configuration options available to you, see the section called “Controlling Access to Kibana” (p. 99), the section called “About Access Policies on VPC Domains” (p. 119), and *Access Control* (p. 30).

Can't Access VPC Domain

See the section called “About Access Policies on VPC Domains” (p. 119) and the section called “Testing VPC Domains” (p. 120).

Red Cluster Status

A red cluster status means that at least one primary shard and its replicas are not allocated to a node. Amazon ES stops taking automatic snapshots, even of healthy indices, while the red cluster status persists.

The most common causes of a red cluster status are failed cluster nodes (p. 146) and the Elasticsearch process crashing due to a continuous heavy processing load.

> **Note**
> Amazon ES stores up to 14 daily automated snapshots for 30 days, so if the red cluster status persists for more than 16 days, permanent data loss can occur. If your Amazon ES domain enters a red cluster status, AWS Support might contact you to ask whether you want to address the problem yourself or you want the support team to assist. You can set a CloudWatch alarm (p. 130) to notify you when a red cluster status occurs.

Ultimately, red shards cause red clusters, and red indices cause red shards. To identity the indices causing the red cluster status, Elasticsearch has some helpful APIs.

- `GET /_cluster/allocation/explain` chooses the first unassigned shard that it finds and explains why it cannot be allocated to a node:

```
{
    "index": "test4",
    "shard": 0,
    "primary": true,
    "current_state": "unassigned",
    "can_allocate": "no",
    "allocate_explanation": "cannot allocate because allocation is not permitted to any
 of the nodes"
}
```

- `GET /_cat/indices?v` shows the health status, number of documents, and disk usage for each index:

```
health status index           uuid                   pri rep docs.count docs.deleted
 store.size pri.store.size
green  open   test1           30h1EiMvS5uAFr2t5CEVoQ   5   0        820            0
   14mb           14mb
green  open   test2           sdIxs_WDT56afFGu5KPbFQ   1   0          0            0
   233b           233b
green  open   test3           GGRZp_TBRZuSaZpAGk2pmw   1   1          2            0
 14.7kb          7.3kb
red    open   test4           BJxfAErbTtu5HBjIXJV_7A   1   0
green  open   test5           _8C6MIXOSxCqVYicH3jsEA   1   0          7            0
 24.3kb         24.3kb
```

Deleting red indices is the fastest way to fix a red cluster status. Depending on the reason for the red cluster status, you might then scale your Amazon ES domain to use larger instance types, more instances, or more EBS-based storage and try to recreate the problematic indices.

If deleting a problematic index isn't feasible, you can restore a snapshot (p. 92), delete documents from the index, change the index settings, reduce the number of replicas, or delete other indices to free up disk space. The important step is to resolve the red cluster status *before* reconfiguring your Amazon ES domain. Reconfiguring a domain with a red cluster status can compound the problem and lead to the domain being stuck in a configuration state of **Processing** until you resolve the status.

Recovering from a Continuous Heavy Processing Load

To determine if a red cluster status is due to a continuous heavy processing load on a data node, monitor the following cluster metrics.

Relevant Metric	Description	Recovery
JVMMemoryPressure	Specifies the percentage of the Java heap used for all data nodes in a cluster. View the **Maximum** statistic for this metric, and look for smaller and smaller drops in memory pressure as the Java garbage collector fails to reclaim sufficient memory. This pattern likely is due to complex queries or large data fields. At 75% memory usage, Elasticsearch triggers the Concurrent Mark Sweep (CMS) garbage collector, which runs alongside other processes to keep pauses and disruptions to a minimum. If CMS fails to reclaim enough memory and usage remains above 75%, Elasticsearch triggers a different garbage collection algorithm that halts	Set memory circuit breakers for the JVM. For more information, see the section called “JVM OutOfMemoryError” (p. 146). If the problem persists, delete unnecessary indices, reduce the number or complexity of requests to the domain, add instances, or use larger instance types.

Relevant Metric	Description	Recovery
	or slows other processes in order to free up sufficient memory to prevent an out of memory error. At 95% memory usage, Elasticsearch kills processes that attempt to allocate memory. It might kill a critical process and bring down one or more nodes in the cluster. The `_nodes/stats/jvm` API offers a useful summary of JVM statistics, memory pool usage, and garbage collection information: `GET elasticsearch_domain/_nodes/stats/jvm?pretty`	
CPUUtilization	Specifies the percentage of CPU resources used for data nodes in a cluster. View the **Maximum** statistic for this metric, and look for a continuous pattern of high usage.	Add data nodes or increase the size of the instance types of existing data nodes. For more information, see the section called "Configuring Amazon ES Domains" (p. 15).
Nodes	Specifies the number of nodes in a cluster. View the **Minimum** statistic for this metric. This value fluctuates when the service deploys a new fleet of instances for a cluster.	Add data nodes. For more information, see the section called "Configuring Amazon ES Domains" (p. 15).

Yellow Cluster Status

A yellow cluster status means that the primary shards for all indices are allocated to nodes in a cluster, but the replica shards for at least one index are not. Single-node clusters always initialize with a yellow cluster status because there is no other node to which Amazon ES can assign a replica. To achieve green cluster status, increase your node count. For more information, see the section called "Sizing Amazon ES Domains" (p. 125) and the section called "Configuring Amazon ES Domains" (p. 15).

ClusterBlockException

You might receive a `ClusterBlockException` error for the following reasons.

Lack of Available Storage Space

If no nodes have enough storage space to accommodate shard relocation, basic write operations like adding documents and creating indices can begin to fail. the section called "Calculating Storage Requirements" (p. 125) provides a summary of how Amazon ES uses disk space.

To avoid issues, monitor the `FreeStorageSpace` metric in the Amazon ES console and create CloudWatch alarms (p. 130) to trigger when `FreeStorageSpace` drops below a certain threshold.

`GET /_cat/allocation?v` also provides a useful summary of shard allocation and disk usage. To resolve issues associated with a lack of storage space, scale your Amazon ES domain to use larger instance types, more instances, or more EBS-based storage. For instructions, see the section called "Configuring Amazon ES Domains" (p. 15).

Block Disks Due to Low Memory

When the **JVMMemoryPressure** metric exceeds 92% for 30 minutes, Amazon ES triggers a protection mechanism and blocks all write operations to prevent the cluster from reaching red status. When the protection is on, write operations fail with a `ClusterBlockException` error, new indices can't be created, and the `IndexCreateBlockException` error is thrown.

When the **JVMMemoryPressure** metric returns to 88% or lower for five minutes, the protection is disabled, and write operations to the cluster are unblocked.

JVM OutOfMemoryError

A JVM `OutOfMemoryError` typically means that one of the following JVM circuit breakers was reached.

Circuit Breaker	Description	Cluster Setting Property
Parent Breaker	Total percentage of JVM heap memory allowed for all circuit breakers. The default value is 70%.	`indices.breaker.total.limit`
Field Data Breaker	Percentage of JVM heap memory allowed to load a single data field into memory. The default value is 60%. If you upload data with large fields, we recommend raising this limit.	`indices.breaker.fielddata.limit`
Request Breaker	Percentage of JVM heap memory allowed for data structures used to respond to a service request. The default value is 40%. If your service requests involve calculating aggregations, we recommend raising this limit.	`indices.breaker.request.limit`

Failed Cluster Nodes

Amazon EC2 instances might experience unexpected terminations and restarts. Typically, Amazon ES restarts the nodes for you. However, it's possible for one or more nodes in an Elasticsearch cluster to remain in a failed condition.

To check for this condition, open your domain dashboard on the Amazon ES console. Choose the **Monitoring** tab, and then choose the **Nodes** metric. See if the reported number of nodes is fewer than the number that you configured for your cluster. If the metric shows that one or more nodes is down for more than one day, contact AWS Support.

You can also set a CloudWatch alarm (p. 130) to notify you when this issue occurs.

> **Note**
> The **Nodes** metric is not accurate during changes to your cluster configuration and during routine maintenance for the service. This behavior is expected. The metric will report the correct number of cluster nodes soon. To learn more, see the section called "About Configuration Changes" (p. 42).

To protect your clusters from unexpected node terminations and restarts, create at least one replica for each index in your Amazon ES domain.

Can't Close Index

Amazon ES doesn't support the `_close` API. If you are restoring an index from a snapshot, you can delete the existing index (before or after reindexing it). The other option is to use the `rename_pattern` and `rename_replacement` fields to rename the index as you restore it:

```
POST /_snapshot/my-repository/my-snapshot/_restore
{
  "indices": "my-index-1,myindex-2",
  "include_global_state": true,
  "rename_pattern": "my-index-(\\d)",
  "rename_replacement": "restored-my-index-$1"
}
```

If you plan to reindex, shrink, or split an index, you likely want to stop writing to it before performing the operation.

Can't SSH into Node

You can't use SSH to access any of the nodes in your Elasticsearch cluster, and you can't directly modify `elasticsearch.yml`. Instead, use the console, AWS CLI, or SDKs to configure your domain. You can specify a few cluster-level settings using the Elasticsearch REST APIs, as well. To learn more, see *Amazon ES Configuration API Reference* (p. 167) and the section called "Supported Elasticsearch Operations" (p. 151).

If you need more insight into the performance of the cluster, you can publish error logs and slow logs to CloudWatch (p. 25).

"Not Valid for the Object's Storage Class" Snapshot Error

Amazon ES snapshots do not support the Amazon Glacier storage class. You might encounter this error when you attempt to list snapshots if your S3 bucket includes a lifecycle rule that transitions objects to the Amazon Glacier storage class.

If you need to restore a snapshot from the bucket, restore the objects from Amazon Glacier, copy the objects to a new bucket, and register the new bucket (p. 89) as a snapshot respository.

Invalid Host Header

Amazon ES requires that clients specify `Host` in the request headers. A valid `Host` value is the domain endpoint without `https://`, such as:

```
Host: search-my-sample-domain-ih2lhn2ew2scurji.us-west-2.es.amazonaws.com
```

If you receive an `Invalid Host Header` error, check that your client includes the Amazon ES domain endpoint (and not, for example, its IP address) in the `Host` header.

Browser Error When Using Kibana

Your browser wraps service error messages in HTTP response objects when you use Kibana to view data in your Amazon ES domain. You can use developer tools commonly available in web browsers, such as Developer Mode in Chrome, to view the underlying service errors and assist your debugging efforts.

To view service errors in Chrome

1. From the menu, choose **View**, **Developer**, **Developer Tools**.
2. Choose the **Network** tab.
3. In the **Status** column, choose any HTTP session with a status of 500.

To view service errors in Firefox

1. From the menu, choose **Tools**, **Web Developer**, **Network**.
2. Choose any HTTP session with a status of 500.
3. Choose the **Response** tab to view the service response.

Unauthorized Operation After Selecting VPC Access

When you create a new domain using the Amazon ES console, you have the option to select VPC or public access. If you select **VPC access**, Amazon ES queries for VPC information and fails if you don't have the proper permissions:

```
You are not authorized to perform this operation. (Service: AmazonEC2; Status Code: 403;
 Error Code: UnauthorizedOperation
```

To enable this query, you must have access to the `ec2:DescribeVpcs`, `ec2:DescribeSubnets`, and `ec2:DescribeSecurityGroups` operations. This requirement is only for the console. If you use the AWS CLI to create and configure a domain with a VPC endpoint, you don't need access to those operations.

Stuck at Loading After Creating VPC Domain

After creating a new domain that uses VPC access, the domain's **Configuration state** might never progress beyond **Loading**. If this issue occurs, you likely have AWS Security Token Service (AWS STS) *disabled* for your region.

To add VPC endpoints to your VPC, Amazon ES needs to assume the `AWSServiceRoleForAmazonElasticsearchService` role. Thus, AWS STS must be enabled to create new domains that use VPC access in a given region. To learn more about enabling and disabling AWS STS, see the IAM User Guide.

Certificate Error When Using SDK

Because AWS SDKs use the CA certificates from your computer, changes to the certificates on the AWS servers can cause connection failures when you attempt to use an SDK. Error messages vary, but typically contain the following text:

```
Failed to query Elasticsearch
...
SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

You can prevent these failures by keeping your computer's CA certificates and operating system up-to-date. If you encounter this issue in a corporate environment and do not manage your own computer, you might need to ask an administrator to assist with the update process.

The following list shows minimum operating system and Java versions:

- Microsoft Windows versions that have updates from January 2005 or later installed contain at least one of the required CAs in their trust list.
- Mac OS X 10.4 with Java for Mac OS X 10.4 Release 5 (February 2007), Mac OS X 10.5 (October 2007), and later versions contain at least one of the required CAs in their trust list.
- Red Hat Enterprise Linux 5 (March 2007), 6, and 7 and CentOS 5, 6, and 7 all contain at least one of the required CAs in their default trusted CA list.
- Java 1.4.2_12 (May 2006), 5 Update 2 (March 2005), and all later versions, including Java 6 (December 2006), 7, and 8, contain at least one of the required CAs in their default trusted CA list.

The three certificate authorities are:

- Amazon Root CA 1
- Starfield Services Root Certificate Authority - G2
- Starfield Class 2 Certification Authority

Root certificates from the first two authorities are available from Amazon Trust Services, but keeping your computer up-to-date is the more straightforward solution. To learn more about ACM-provided certificates, see AWS Certificate Manager FAQs.

Note
Currently, Amazon ES domains in the us-east-1 region use certificates from a different authority. We plan to update the region to use these new certificate authorities in the near future.

Amazon Elasticsearch Service General Reference

Amazon Elasticsearch Service (Amazon ES) supports a variety of instances, operations, plugins, and other resources.

Topics

Supported Instance Types

Amazon ES supports the following instance types. Not all regions support all instance types. For availability details, see Amazon Elasticsearch Service Pricing.

For information about which instance type is appropriate for your use case, see the section called "Sizing Amazon ES Domains" (p. 125), the section called "EBS Volume Size Limits" (p. 225), and the section called "Network Limits" (p. 227).

Instance Type	Restrictions
C4	
I2	
I3	The I3 instance types do not support EBS storage volumes and require Elasticsearch version 5.1 or newer.
M3	The M3 instance types do not support encryption of data at rest.
M4	
R3	The R3 instance types do not support encryption of data at rest.
R4	
T2	• You can use the T2 instance types only if the instance count for your domain is 10 or fewer. • The `t2.micro.elasticsearch` instance type supports only Elasticsearch 2.3 and 1.5. • The T2 instance types do not support encryption of data at rest.

Tip
You can use different instance types for dedicated master nodes (p. 128) and data nodes.

Supported Elasticsearch Operations

Amazon ES supports many versions of Elasticsearch. The following topics show the operations that Amazon ES supports for each version.

Topics

Notable API Differences

Cluster Settings

Prior to Elasticsearch 5.3, the `_cluster/settings` API on Amazon ES domains supported only the HTTP `PUT` method, not the `GET` method. Newer versions support the `GET` method, as shown in the following example:

```
GET https://domain.region.es.amazonaws.com/_cluster/settings?pretty
```

A sample return follows:

```
{
  "persistent" : {
    "cluster" : {
      "routing" : {
        "allocation" : {
          "cluster_concurrent_rebalance" : "2"
        }
      }
    },
    "indices" : {
      "recovery" : {
        "max_bytes_per_sec" : "20mb"
      }
    }
  },
  "transient" : {
    "cluster" : {
      "routing" : {
        "allocation" : {
          "exclude" : {
            "di_number" : "2"
          }
        }
      }
    }
  }
}
```

- `cluster_concurrent_rebalance` specifies the number of shards that can be relocated to new nodes at any given time.
- `max_bytes_per_sec` is the maximum data transfer speed that Elasticsearch uses during a recovery event.
- `di_number` is an internal Amazon ES value that is used to copy shards to new *domain instances* after configuration changes.

Shrink

The `_shrink` API can cause upgrades, configuration changes, and domain deletions to fail. We don't recommend using it on domains that run Elasticsearch versions 5.3 or 5.1. These versions have a bug that can cause snapshot restoration of shrunken indices to fail.

If you use the `_shrink` API on other Elasticsearch versions, make the following request before starting the shrink operation:

```
PUT https://domain.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": "name-of-the-node-to-shrink-to",
    "index.blocks.read_only": true
  }
}
```

Then make the following request after completing the shrink operation:

```
PUT https://domain.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}
```

Version 6.3

For Elasticsearch 6.3, Amazon ES supports the following operations.

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cache/clear` (Index only)
- `/_cat`
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` for several properties[4]:
- `/_delete_by_query`[1]
- `/_explain`
- `/_field_stats`
- `/_flush`
- `/_forcemerge` (Index only)
- `/_ingest`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugin/kibana`
- `/_scripts`[3]

  `/_search`[2]
- `/_search profile`
- `/_segments` (Index only)
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`

  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`

- `/_rank_eval`
- `/_recovery` (Index only)
- `/_refresh`
- `/_reindex`[1]
- `/_rollover`

- `/_termvectors` (Index only)
- `/_update`[3]
- `/_update_by_query`[1]
- `/_validate`

1. Cluster configuration changes might interrupt these operations before completion. We recommend that you use the `/_tasks` operation along with these operations to verify that the requests completed successfully.
2. DELETE requests to `/_search/scroll` with a message body must specify `"Content-Length"` in the HTTP header. Most clients add this header by default. To avoid a problem with = characters in `scroll_id` values, use the request body, not the query string, to pass `scroll_id` values to Amazon ES.
3. For considerations about using scripts, see the section called "Other Supported Resources" (p. 161).
4. Refers to the `PUT` method. For information about the `GET` method, see the section called "Notable API Differences" (p. 151).
5. See the section called "Shrink" (p. 152).

Version 6.2

For Elasticsearch 6.2, Amazon ES supports the following operations.

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cache/clear` (Index only)
- `/_cat`
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` for several properties[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`

- `/_delete_by_query`[1]
- `/_explain`
- `/_field_stats`
- `/_flush`
- `/_forcemerge` (Index only)
- `/_ingest`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugin/kibana`
- `/_rank_eval`
- `/_recovery` (Index only)
- `/_refresh`
- `/_reindex`[1]
- `/_rollover`

- `/_scripts`[3]

  `/_search`[2]
- `/_search profile`
- `/_segments` (Index only)
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_termvectors` (Index only)
- `/_update`[3]
- `/_update_by_query`[1]
- `/_validate`

[1] Cluster configuration changes might interrupt these operations before completion. We recommend that you use the `/_tasks` operation along with these operations to verify that the requests completed successfully.

[2] DELETE requests to `/_search/scroll` with a message body must specify `"Content-Length"` in the HTTP header. Most clients add this header by default. To avoid a problem with = characters in `scroll_id` values, use the request body, not the query string, to pass `scroll_id` values to Amazon ES.

[3] For considerations about using scripts, see the section called “Other Supported Resources” (p. 161).

[4] Refers to the `PUT` method. For information about the `GET` method, see the section called “Notable API Differences” (p. 151).

[5] See the section called “Shrink” (p. 152).

Version 6.0

For Elasticsearch 6.0, Amazon ES supports the following operations.

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cache/clear` (Index only)
- `/_cat`
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` for several properties[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`

- `/_delete_by_query`[1]
- `/_explain`
- `/_field_stats`
- `/_flush`
- `/_forcemerge` (Index only)
- `/_ingest`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugin/kibana`
- `/_recovery` (Index only)
- `/_refresh`
- `/_reindex`[1]
- `/_rollover`

- `/_scripts`[3]

  `/_search`[2]
- `/_search profile`
- `/_segments` (Index only)
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_termvectors` (Index only)
- `/_update`[3]
- `/_update_by_query`[1]
- `/_validate`

[1] Cluster configuration changes might interrupt these operations before completion. We recommend that you use the `/_tasks` operation along with these operations to verify that the requests completed successfully.

[2] DELETE requests to `/_search/scroll` with a message body must specify `"Content-Length"` in the HTTP header. Most clients add this header by default. To avoid a problem with = characters in `scroll_id` values, use the request body, not the query string, to pass `scroll_id` values to Amazon ES.

[3] For considerations about using scripts, see the section called “Other Supported Resources” (p. 161).

[4] Refers to the `PUT` method. For information about the `GET` method, see the section called “Notable API Differences” (p. 151).

5. See the section called “Shrink” (p. 152).

Version 5.6

For Elasticsearch 5.6, Amazon ES supports the following operations.

- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cache/clear (Index only)
- /_cat
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings for several properties[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count

- /_delete_by_query[1]
- /_explain
- /_field_stats
- /_flush
- /_forcemerge (Index only)
- /_ingest
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugin/kibana
- /_recovery (Index only)
- /_refresh
- /_reindex[1]
- /_rollover

- /_scripts[3]

  /_search[2]
- /_search profile
- /_segments (Index only)
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_termvectors (Index only)
- /_update[3]
- /_update_by_query[1]
- /_validate

1. Cluster configuration changes might interrupt these operations before completion. We recommend that you use the /_tasks operation along with these operations to verify that the requests completed successfully.
2. DELETE requests to /_search/scroll with a message body must specify "Content-Length" in the HTTP header. Most clients add this header by default. To avoid a problem with = characters in scroll_id values, use the request body, not the query string, to pass scroll_id values to Amazon ES.
3. For considerations about using scripts, see the section called “Other Supported Resources” (p. 161).
4. Refers to the PUT method. For information about the GET method, see the section called “Notable API Differences” (p. 151).
5. See the section called “Shrink” (p. 152).

Version 5.5

For Elasticsearch 5.5, Amazon ES supports the following operations.

- /_alias
- /_aliases

- /_delete_by_query[1]
- /_explain

- /_scripts[3]

- /_all
- /_analyze
- /_bulk
- /_cache/clear (Index only)
- /_cat
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings for several properties[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_field_stats
- /_flush
- /_forcemerge (Index only)
- /_ingest
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugin/kibana
- /_recovery (Index only)
- /_refresh
- /_reindex[1]
- /_rollover

/_search[2]

- /_search profile
- /_segments (Index only)
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_termvectors (Index only)
- /_update[3]
- /_update_by_query[1]
- /_validate

1. Cluster configuration changes might interrupt these operations before completion. We recommend that you use the /_tasks operation along with these operations to verify that the requests completed successfully.
2. DELETE requests to /_search/scroll with a message body must specify "Content-Length" in the HTTP header. Most clients add this header by default. To avoid a problem with = characters in scroll_id values, use the request body, not the query string, to pass scroll_id values to Amazon ES.
3. For considerations about using scripts, see the section called “Other Supported Resources” (p. 161).
4. Refers to the PUT method. For information about the GET method, see the section called “Notable API Differences” (p. 151).
5. See the section called “Shrink” (p. 152).

Version 5.3

For Elasticsearch 5.3, Amazon ES supports the following operations.

- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cache/clear (Index only)
- /_cat
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_delete_by_query[1]
- /_explain
- /_field_stats
- /_flush
- /_forcemerge (Index only)
- /_ingest
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_search[2]
- /_search profile
- /_segments (Index only)
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template

- `/_cluster/settings` for several properties[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_nodes`
- `/_plugin/kibana`
- `/_recovery` (Index only)
- `/_refresh`
- `/_reindex`[1]
- `/_rollover`
- `/_termvectors` (Index only)
- `/_update`[3]
- `/_update_by_query`[1]
- `/_validate`

1. Cluster configuration changes might interrupt these operations before completion. We recommend that you use the `/_tasks` operation along with these operations to verify that the requests completed successfully.
2. DELETE requests to `/_search/scroll` with a message body must specify `"Content-Length"` in the HTTP header. Most clients add this header by default. To avoid a problem with = characters in `scroll_id` values, use the request body, not the query string, to pass `scroll_id` values to Amazon ES.
3. For considerations about using scripts, see the section called “Other Supported Resources” (p. 161).
4. Refers to the `PUT` method. For information about the `GET` method, see the section called “Notable API Differences” (p. 151).
5. See the section called “Shrink” (p. 152).

Version 5.1

For Elasticsearch 5.1, Amazon ES supports the following operations.

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cache/clear` (Index only)
- `/_cat`
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` for several properties (PUT only):
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_cluster/state`
- `/_count`
- `/_delete_by_query`[1]
- `/_explain`
- `/_field_stats`
- `/_flush`
- `/_forcemerge` (Index only)
- `/_ingest`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugin/kibana`
- `/_recovery` (Index only)
- `/_refresh`
- `/_reindex`[1]
- `/_rollover`
- `/_search`[2]
- `/_search profile`
- `/_segments` (Index only)
- `/_shard_stores`
- `/_shrink`[4]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_termvectors` (Index only)
- `/_update`[3]
- `/_update_by_query`[1]
- `/_validate`

- `/_cluster/stats`

1. Cluster configuration changes might interrupt these operations before completion. We recommend that you use the `/_tasks` operation along with these operations to verify that the requests completed successfully.
2. DELETE requests to `/_search/scroll` with a message body must specify `"Content-Length"` in the HTTP header. Most clients add this header by default. To avoid a problem with = characters in `scroll_id` values, use the request body, not the query string, to pass `scroll_id` values to Amazon ES.
3. For considerations about using scripts, see the section called “Other Supported Resources” (p. 161).
4. See the section called “Shrink” (p. 152).

Version 2.3

For Elasticsearch 2.3, Amazon ES supports the following operations.

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cache/clear` (Index only)
- `/_cat`
- `/_cluster/health`
- `/_cluster/settings` for four properties (PUT only):
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
  - `threadpool.bulk.queue_size`
- `/_cluster/stats`
- `/_count`
- `/_flush`
- `/_forcemerge` (Index only)
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_nodes`
- `/_percolate`
- `/_plugin/kibana`
- `/_recovery` (Index only)
- `/_refresh`
- `/_search`
- `/_segments` (Index only)
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_template`

Version 1.5

For Elasticsearch 1.5, Amazon ES supports the following operations.

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`
- `/_cluster/health`
- `/_cluster/settings` for four properties (PUT only):
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
- `/_cluster/stats`
- `/_count`
- `/_flush`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_nodes`
- `/_percolate`
- `/_plugin/kibana`
- `/_plugin/kibana3`
- `/_refresh`
- `/_search`
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_template`

• `indices.breaker.total.limit` • `threadpool.bulk.queue_size`		

Supported Plugins

Amazon ES domains come prepackaged with plugins that are available from the Elasticsearch community. The service automatically deploys and manages plugins for you.

> **Note**
> Kibana is a plugin in older versions of Amazon ES and a Node.js application in newer versions. All Amazon ES domains include a preinstalled version of Kibana.

Elasticsearch Version	Plugins
6.3	• ICU Analysis • Ingest Attachment Processor • Ingest User Agent Processor • Japanese (kuromoji) Analysis • Mapper Murmur3 • Mapper Size • Phonetic Analysis • Smart Chinese Analysis • Stempel Polish Analysis • Ukrainian Analysis • Seunjeon Korean Analysis
6.2	• ICU Analysis • Ingest Attachment Processor • Ingest User Agent Processor • Japanese (kuromoji) Analysis • Mapper Murmur3 • Mapper Size • Phonetic Analysis • Smart Chinese Analysis • Stempel Polish Analysis • Ukrainian Analysis • Seunjeon Korean Analysis
6.0	• ICU Analysis • Ingest Attachment Processor • Ingest User Agent Processor • Japanese (kuromoji) Analysis • Mapper Murmur3 • Mapper Size • Phonetic Analysis • Smart Chinese Analysis • Stempel Polish Analysis • Ukrainian Analysis

Elasticsearch Version	Plugins
	• Seunjeon Korean Analysis
5.6	• ICU Analysis • Ingest Attachment Processor • Ingest User Agent Processor • Japanese (kuromoji) Analysis • Mapper Attachments • Mapper Murmur3 • Mapper Size • Phonetic Analysis • Smart Chinese Analysis • Stempel Polish Analysis • Ukrainian Analysis • Seunjeon Korean Analysis
5.5	• ICU Analysis • Ingest Attachment Processor • Ingest User Agent Processor • Japanese (kuromoji) Analysis • Mapper Attachments • Mapper Murmur3 • Mapper Size • Phonetic Analysis • Smart Chinese Analysis • Stempel Polish Analysis • Ukrainian Analysis • Seunjeon Korean Analysis
5.3	• ICU Analysis • Ingest Attachment Processor • Ingest User Agent Processor • Japanese (kuromoji) Analysis • Mapper Attachments • Mapper Murmur3 • Mapper Size • Phonetic Analysis • Smart Chinese Analysis • Stempel Polish Analysis • Ukrainian Analysis • Seunjeon Korean Analysis

Elasticsearch Version	Plugins
5.1	• ICU Analysis • Ingest Attachment Processor • Ingest User Agent Processor • Japanese (kuromoji) Analysis • Mapper Murmur3 • Phonetic Analysis • Smart Chinese Analysis • Stempel Polish Analysis • Seunjeon Korean Analysis
2.3	• ICU Analysis • Japanese (kuromoji) Analysis • Kibana 4 • Phonetic Analysis
1.5	• ICU Analysis • Japanese (kuromoji) Analysis • Kibana 3 (at the `_plugin/kibana3/` endpoint) • Kibana 4

Output Plugins

Amazon ES supports two Logstash output plugins to stream data into Amazon ES: the standard Elasticsearch output plugin and the logstash-output-amazon-es plugin, which signs and exports Logstash events to Amazon ES.

For more information about Logstash, see the section called “Loading Bulk Data with the Logstash Plugin” (p. 102).

Other Supported Resources

bootstrap.mlockall

The service enables `bootstrap.mlockall` in `elasticsearch.yml`, which locks JVM memory and prevents the operating system from swapping it to disk. This applies to all supported instance types except for the following:

- `t2.micro.elasticsearch`
- `t2.small.elasticsearch`
- `t2.medium.elasticsearch`

Scripting module

The service supports scripting for Elasticsearch 5.*x* and newer domains. The service does not support scripting for 1.5 or 2.3.

Supported scripting options include the following:

- Painless

- Lucene Expressions
- Mustache

For Elasticsearch 5.5 and newer domains, Amazon ES supports stored scripts using the `_scripts` endpoint. Elasticsearch 5.3 and 5.1 domains only support inline scripts.

TCP transport

The service supports HTTP on port 80, but does not support TCP transport.

Using the AWS SDKs with Amazon Elasticsearch Service

This chapter includes examples of how to use the AWS SDKs to interact with the Amazon Elasticsearch Service Configuration API. These code samples show how to create, update, and delete Amazon ES domains.

> **Important**
> For examples of how to interact with the Elasticsearch APIs, such as `_index`, `_bulk`, `_search`, and `_snapshot`, see *Signing HTTP Requests* (p. 62).

Java

This first example uses the AWS SDK for Java to create a domain, update its configuration, and delete it. Uncomment the calls to `waitForDomainProcessing` (and comment the call to `deleteDomain`) to allow the domain to come online and be useable.

```
package com.amazonaws.samples;

import java.util.concurrent.TimeUnit;

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticsearch.AWSElasticsearch;
import com.amazonaws.services.elasticsearch.AWSElasticsearchClientBuilder;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.EBSOptions;
import com.amazonaws.services.elasticsearch.model.ElasticsearchClusterConfig;
import com.amazonaws.services.elasticsearch.model.NodeToNodeEncryptionOptions;
import com.amazonaws.services.elasticsearch.model.ResourceNotFoundException;
import com.amazonaws.services.elasticsearch.model.SnapshotOptions;
import com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigRequest;
import com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigResult;
import com.amazonaws.services.elasticsearch.model.VolumeType;

/**
 * Sample class demonstrating how to use the AWS SDK for Java to create, update,
 * and delete Amazon Elasticsearch Service domains.
 */

public class AESSample {

    public static void main(String[] args) {

        final String domainName = "my-test-domain";

        // Build the client using the default credentials chain.
        // You can use the AWS CLI and run `aws configure` to set access key, secret
        // key, and default region.
        final AWSElasticsearch client = AWSElasticsearchClientBuilder
                .standard()
                // Unnecessary, but lets you use a region different than your default.
                .withRegion(Regions.US_WEST_2)
                // Unnecessary, but if desired, you can use a different provider chain.
                .withCredentials(new DefaultAWSCredentialsProviderChain())
                .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon Elasticsearch Service domain with the specified options.
     * Some options require other AWS resources, such as an Amazon Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The AWSElasticsearch client to use for the requests to Amazon
     *            Elasticsearch Service
     * @param domainName
     *            The name of the domain you want to create
     */
    private static void createDomain(final AWSElasticsearch client, final String
 domainName) {

        // Create the request and set the desired configuration options
        CreateElasticsearchDomainRequest createRequest = new
 CreateElasticsearchDomainRequest()
                .withDomainName(domainName)
                .withElasticsearchVersion("6.3")
                .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                        .withDedicatedMasterEnabled(true)
                        .withDedicatedMasterCount(3)
                        // Small, inexpensive instance types for testing. Not recommended
 for production
                        // domains.
                        .withDedicatedMasterType("t2.small.elasticsearch")
                        .withInstanceType("t2.small.elasticsearch")
                        .withInstanceCount(5))
                // Many instance types require EBS storage.
                .withEBSOptions(new EBSOptions()
                        .withEBSEnabled(true)
                        .withVolumeSize(10)
                        .withVolumeType(VolumeType.Gp2))
                // You can uncomment this line and add your account ID, a user name, and
 the
                // domain name to add an access policy.
                // .withAccessPolicies("{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name
\"]},\"Action\":[\"es:*\"],\"Resource\":\"arn:aws:es:region:123456789012:domain/domain-
name/*\"}]}")
                .withNodeToNodeEncryptionOptions(new NodeToNodeEncryptionOptions()
                        .withEnabled(true));

        // Make the request.
        System.out.println("Sending domain creation request...");
        CreateElasticsearchDomainResult createResponse =
 client.createElasticsearchDomain(createRequest);
        System.out.println("Domain creation response from Amazon Elasticsearch Service:");
        System.out.println(createResponse.getDomainStatus().toString());
```

```
   }

   /**
    * Updates the configuration of an Amazon Elasticsearch Service domain with the
    * specified options. Some options require other AWS resources, such as an
    * Amazon Cognito user pool and identity pool, whereas others require just an
    * instance type or instance count.
    *
    * @param client
    *            The AWSElasticsearch client to use for the requests to Amazon
    *            Elasticsearch Service
    * @param domainName
    *            The name of the domain to update
    */
   private static void updateDomain(final AWSElasticsearch client, final String
domainName) {
       try {
           // Updates the domain to take automated snapshots at noon and use three data
           // instances instead of five.
           // You can uncomment the Cognito lines and fill in the strings to enable
Cognito
           // authentication for Kibana.
           final UpdateElasticsearchDomainConfigRequest updateRequest = new
UpdateElasticsearchDomainConfigRequest()
                   .withDomainName(domainName)
                   .withSnapshotOptions(new SnapshotOptions()
                           .withAutomatedSnapshotStartHour(12))
                   // .withCognitoOptions(new CognitoOptions()
                           // .withEnabled(true)
                           // .withUserPoolId("user-pool-id")
                           // .withIdentityPoolId("identity-pool-id")
                           // .withRoleArn("role-arn"))
                   .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                           .withInstanceCount(3));

           System.out.println("Sending domain update request...");
           final UpdateElasticsearchDomainConfigResult updateResponse = client
                   .updateElasticsearchDomainConfig(updateRequest);
           System.out.println("Domain update response from Amazon Elasticsearch
Service:");
           System.out.println(updateResponse.toString());
       } catch (ResourceNotFoundException e) {
           System.out.println("Domain not found. Please check the domain name.");
       }
   }

   /**
    * Deletes an Amazon Elasticsearch Service domain. Deleting a domain can take
    * several minutes.
    *
    * @param client
    *            The AWSElasticsearch client to use for the requests to Amazon
    *            Elasticsearch Service
    * @param domainName
    *            The name of the domain that you want to delete
    */
   private static void deleteDomain(final AWSElasticsearch client, final String
domainName) {
       try {
           final DeleteElasticsearchDomainRequest deleteRequest = new
DeleteElasticsearchDomainRequest()
                   .withDomainName(domainName);

           System.out.println("Sending domain deletion request...");
           final DeleteElasticsearchDomainResult deleteResponse =
client.deleteElasticsearchDomain(deleteRequest);
```

```
            System.out.println("Domain deletion response from Amazon Elasticsearch
 Service:");
            System.out.println(deleteResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }

    /**
     * Waits for the domain to finish processing changes. New domains typically take
     * 10-15 minutes to initialize. Most updates to existing domains take a similar
     * amount of time. This method checks every 15 seconds and finishes only when
     * the domain's processing status changes to false.
     *
     * @param client
     *            The AWSElasticsearch client to use for the requests to Amazon
     *            Elasticsearch Service
     * @param domainName
     *            The name of the domain that you want to check
     */
    private static void waitForDomainProcessing(final AWSElasticsearch client, final String
 domainName) {
        // Create a new request to check the domain status.
        final DescribeElasticsearchDomainRequest describeRequest = new
 DescribeElasticsearchDomainRequest()
                .withDomainName(domainName);

        // Check whether the domain is processing, which usually takes 10-15 minutes
        // after creation or a configuration change.
        // This loop checks every 15 seconds.
        DescribeElasticsearchDomainResult describeResponse =
 client.describeElasticsearchDomain(describeRequest);
        while (describeResponse.getDomainStatus().isProcessing()) {
            try {
                System.out.println("Domain still processing...");
                TimeUnit.SECONDS.sleep(15);
                describeResponse = client.describeElasticsearchDomain(describeRequest);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        // Once we exit that loop, the domain is available
        System.out.println("Amazon Elasticsearch Service has finished processing changes
 for your domain.");
        System.out.println("Domain description response from Amazon Elasticsearch
 Service:");
        System.out.println(describeResponse.toString());
    }
}
```

Amazon Elasticsearch Service Configuration API Reference

This reference describes the actions, data types, and errors in the Amazon Elasticsearch Service Configuration API. The Configuration API is a REST API that you can use to create and configure Amazon ES domains over HTTP. You also can use the AWS CLI and the console to configure Amazon ES domains. For more information, see Creating and Configuring Amazon ES Domains (p. 10).

- Actions (p. 167)
- Data Types (p. 210)
- Errors (p. 223)

Actions

The following table provides a quick reference to the HTTP method required for each operation for the REST interface to the Amazon Elasticsearch Service Configuration API. The description of each operation also includes the required HTTP method.

Note
All configuration service requests must be signed. For more information, see Signing Amazon Elasticsearch Service Requests (p. 33) in this guide and Signature Version 4 Signing Process in the *AWS General Reference*.

Action	HTTP Method
`AddTags` (p. 168)	`POST`
`CreateElasticsearchDomain` (p. 169)	`POST`
`DeleteElasticsearchDomain` (p. 174)	`DELETE`
`DeleteElasticsearchServiceRole` (p. 176)	`DELETE`
`DescribeElasticsearchDomain` (p. 176)	`GET`
`DescribeElasticsearchDomainConfig` (p. 178)	`GET`
`DescribeElasticsearchDomains` (p. 181)	`POST`
`DescribeElasticsearchInstanceTypeLimits` (p. 184)	`GET`
`DescribeReservedElasticsearchInstanceOfferings` (p. 187)	`GET`
`DescribeReservedElasticsearchInstances` (p. 188)	`GET`
`GetCompatibleElasticsearchVersions` (p. 190)	`GET`
`GetUpgradeHistory` (p. 192)	`GET`
`GetUpgradeStatus` (p. 194)	`GET`
`ListDomainNames` (p. 195)	`GET`

Action	HTTP Method
ListElasticsearchInstanceTypeDetails (p. 196)	GET
ListElasticsearchInstanceTypes (p. 198)	GET
ListElasticsearchVersions (p. 199)	GET
ListTags (p. 201)	GET
PurchaseReservedElasticsearchInstance (p. 202)	POST
RemoveTags (p. 203)	POST
UpdateElasticsearchDomainConfig (p. 204)	POST
UpgradeElasticsearchDomain (p. 209)	POST

AddTags

Attaches resource tags to an Amazon ES domain. For more information, see Tagging Amazon ES Domains (p. 55).

Syntax

```
POST /2015-01-01/tags
{
    "ARN": "<DOMAIN_ARN>",
    "TagList": [
        {
            "Key": "<TAG_KEY>",
            "Value": "<TAG_VALUE>"
        }
    ]
}
```

Request Parameters

This operation does not use request parameters.

Request Body

Parameter	Data Type	Required?	Description
TagList	TagList (p. 222)	Yes	List of resource tags
ARN	ARN (p. 211)	Yes	Amazon Resource Name (ARN) for the Amazon ES domain to which you want to attach resource tags.

Response Elements

Not applicable. The AddTags operation does not return a data structure.

Errors

The `AddTags` operation can return any of the following errors:

- `BaseException` (p.)
- `LimitExceededException` (p.)
- `ValidationException` (p.)
- `InternalException` (p.)

Example

The following example attaches a single resource tag with a tag key of `project` to the `logs` Amazon ES domain:

Request

```
POST es.<AWS_REGION>.amazonaws.com/2015-01-01/tags
{
    "ARN": "<DOMAIN_ARN>",
    "TagList": [
        {
            "Key": "project",
            "Value": "trident"
        }
    ]
}
```

Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 5a6a5790-536c-11e5-9cd2-b36dbf43d89e
Content-Type: application/json
Content-Length: 0
Date: Sat, 05 Sep 2015 01:20:55 GMT
```

CreateElasticsearchDomain

Creates a new Amazon ES domain. For more information, see the section called " Creating Amazon ES Domains" (p. 10).

> **Note**
> If you attempt to create an Amazon ES domain and a domain with the same name already exists, the API does not report an error. Instead, it returns details for the existing domain.

Syntax

```
POST /2015-01-01/es/domain
{
    "DomainName": "<DOMAIN_NAME>",
    "ElasticsearchVersion": "<VERSION>",
    "ElasticsearchClusterConfig": {
        "InstanceType": "<INSTANCE_TYPE>",
        "InstanceCount": <INSTANCE_COUNT>,
        "DedicatedMasterEnabled": "<TRUE|FALSE>",
        "DedicatedMasterCount": <INSTANCE_COUNT>,
        "DedicatedMasterType": "<INSTANCE_TYPE>",
        "ZoneAwarenessEnabled": "<TRUE|FALSE>"
    },
    "EBSOptions": {
        "EBSEnabled": "<TRUE|FALSE>",
        "VolumeType": "<VOLUME_TYPE>",
        "VolumeSize": "<VOLUME_SIZE>",
        "Iops": "<VALUE>"
    },
    "VPCOptions": {
        "SubnetIds": [
            "<SUBNET_ID>"
        ],
        "SecurityGroupIds": [
            "<SECURITY_GROUP_ID>"
        ]
    },
    "CognitoOptions": {
        "IdentityPoolId": "us-west-1:12345678-1234-1234-1234-123456789012",
        "RoleArn": "arn:aws:iam::123456789012:role/my-kibana-role",
        "Enabled": true,
        "UserPoolId": "us-west-1_121234567"
    },
    "AccessPolicies": "<ACCESS_POLICY_DOCUMENT>",
    "SnapshotOptions": {
        "AutomatedSnapshotStartHour": <START_HOUR>
    },
    "LogPublishingOptions": {
        "SEARCH_SLOW_LOGS": {
            "CloudWatchLogsLogGroupArn":"<ARN>",
            "Enabled":true
        },
        "INDEX_SLOW_LOGS": {
            "CloudWatchLogsLogGroupArn":"<ARN>",
            "Enabled":true
        }
    },
    "EncryptionAtRestOptions": {
        "Enabled": true,
        "KmsKeyId": "<KEY_ID>"
    },
    "AdvancedOptions": {
        "rest.action.multi.allow_explicit_index": "<TRUE|FALSE>",
        "indices.fielddata.cache.size": "<PERCENTAGE_OF_HEAP>"
    },
    "NodeToNodeEncryptionOptions": {
        "Enabled": true|false
    }
}
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Parameter	Data Type	Required?	Description
DomainName	DomainName (p. 213)	Yes	Name of the Amazon ES domain to create.
ElasticsearchVersion	String	No	Version of Elasticsearch. If not specified, 1.5 is used as the default. For the full list of supported versions, see

Parameter	Data Type	Required?	Description
			the section called “Supported Elasticsearch Versions” (p. 2).
ElasticsearchClusterConfig	ElasticsearchClusterConfig (p. 214)	No	Container for the cluster configuration of an Amazon ES domain.
EBSOptions	EBSOptions (p. 214)	No	Container for the parameters required to enable EBS-based storage for an Amazon ES domain. For more information, see Configuring EBS-based Storage (p. 18).
VPCOptions	VPCOptions (p. 223)	No	Container for the values required to configure VPC access domains. If you don't specify these values, Amazon ES creates the domain with a public endpoint. To learn more, see VPC Support for Amazon Elasticsearch Service Domains (p. 117).
CognitoOptions	CognitoOptions (p. 212)	No	Key-value pairs to configure Amazon ES to use Amazon Cognito authentication for Kibana.
AccessPolicies	String	No	IAM policy document specifying the access policies for the new Amazon ES domain. For more information, see *Access Control* (p. 30).
SnapshotOptions	SnapshotOptions (p. 221)	No	Container for parameters required to configure automated snapshots of domain indices. For more information, see Configuring Snapshots (p. 23).
AdvancedOptions	AdvancedOptions (p. 210)	No	Key-value pairs to specify advanced configuration options. For more information, see Configuring Advanced Options (p. 24).
LogPublishingOptions	LogPublishingOptions (p. 218)	No	Key-value pairs to configure slow log publishing.
EncryptionAtRestOptions	EncryptionAtRestOptions (p. 218)	No	Key-value pairs to enable encryption at rest.
NodeToNodeEncryptionOptions	NodeToNodeEncryptionOptions (p. 219)	No	Enables node-to-node encryption.

Response Elements

Field	Data Type	Description
`DomainStatus`	ElasticsearchDomainStatus (p. 215)	Specifies the status and configuration of a new Amazon ES domain.

Errors

`CreateElasticsearchDomain` can return any of the following errors:

- `BaseException` (p.)
- `DisabledOperationException` (p.)
- `InternalException` (p.)
- `InvalidTypeException` (p.)
- `LimitExceededException` (p.)
- `ResourceAlreadyExistsException` (p.)
- `ValidationException` (p.)

Example

This example demonstrates the following:

- Creates an Amazon ES domain that is named `streaming-logs`
- Configures a cluster with six data nodes (`i3.large`) and three dedicated master nodes (`c4.large`)
- Enables zone awareness
- Configures VPC access for the domain
- Enables encryption at rest and node-to-node encryption

Request

```
POST https://es.us-west-1.amazonaws.com/2015-01-01/es/domain
{
  "DomainName": "streaming-logs",
  "ElasticsearchVersion": "6.3",
  "ElasticsearchClusterConfig": {
    "InstanceType": "i3.large.elasticsearch",
    "InstanceCount": 6,
    "DedicatedMasterEnabled": "true",
    "DedicatedMasterCount": 3,
    "DedicatedMasterType": "c4.large.elasticsearch",
    "ZoneAwarenessEnabled": "true"
  },
  "EncryptionAtRestOptions": {
    "Enabled": true,
    "KmsKeyId": "1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "VPCOptions": {
    "SubnetIds": [
      "subnet-87654321",
      "subnet-12345678"
    ]
  }
}
```

Response

```
{
  "DomainStatus": {
    "ARN": "arn:aws:es:us-west-1:123456789012:domain/streaming-logs",
    "AccessPolicies": "",
    "AdvancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    },
    "CognitoOptions": {
      "Enabled": false,
      "IdentityPoolId": null,
      "RoleArn": null,
      "UserPoolId": null
    },
    "Created": true,
    "Deleted": false,
    "DomainId": "123456789012/streaming-logs",
    "DomainName": "streaming-logs",
    "EBSOptions": {
      "EBSEnabled": false,
      "Iops": null,
      "VolumeSize": null,
      "VolumeType": null
    },
    "ElasticsearchClusterConfig": {
      "DedicatedMasterCount": 3,
      "DedicatedMasterEnabled": true,
      "DedicatedMasterType": "c4.large.elasticsearch",
      "InstanceCount": 6,
      "InstanceType": "i3.large.elasticsearch",
      "ZoneAwarenessEnabled": true
    },
    "ElasticsearchVersion": "6.3",
    "EncryptionAtRestOptions": {
      "Enabled": true,
      "KmsKeyId": "arn:aws:kms:us-
west-1:123456789012:key/1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
    },
    "Endpoint": null,
    "Endpoints": null,
    "LogPublishingOptions": null,
    "NodeToNodeEncryptionOptions": {
      "Enabled": true
    },
    "Processing": true,
    "ServiceSoftwareOptions": {
      "AutomatedUpdateDate": 0,
      "Cancellable": false,
      "CurrentVersion": "LEGACY",
      "Description": "There is no software update available for this domain.",
      "NewVersion": "",
      "UpdateAvailable": false,
      "UpdateStatus": "COMPLETED"
    },
    "SnapshotOptions": {
      "AutomatedSnapshotStartHour": 0
    },
    "UpgradeProcessing": false,
    "VPCOptions": {
      "AvailabilityZones": [
        "us-west-1b",
        "us-west-1c"
      ],
      "SecurityGroupIds": [
        "sg-12345678"
      ],
      "SubnetIds": [
        "subnet-12345678",
        "subnet-87654321"
      ],
      "VPCId": "vpc-12345678"
    }
  }
}
```

DeleteElasticsearchDomain

Deletes an Amazon ES domain and all of its data. A domain cannot be recovered after it is deleted.

Syntax

```
DELETE /2015-01-01/es/domain/<DOMAIN_NAME>
```

Request Parameters

Parameter	Data Type	Required?	Description
`DomainName`	`DomainName` (p. 213)	Yes	Name of the Amazon ES domain that you want to delete.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
`DomainStatus`	ElasticsearchDomainStatus (p. 215)	Specifies the configuration of the specified Amazon ES domain.

Errors

The `DeleteElasticsearchDomain` operation can return any of the following errors:

- `BaseException` (p.)
- `InternalException` (p.)
- `ResourceNotFoundException` (p.)
- `ValidationException` (p.)

Example

The following example deletes the `weblogs` domain:

Request

```
DELETE es.<AWS_REGION>.amazonaws.com/2015-01-01/es/domain/weblogs
```

Response

```
HTTP/1.1 200 OK
{
    "DomainStatus": {
        "ARN": "arn:aws:es:us-west-1:123456789012:domain/weblogs",
        "AccessPolicies": "",
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "Created": true,
        "Deleted": true,
        "DomainId": "123456789012/weblogs",
        "DomainName": "weblogs",
        "EBSOptions": {
            "EBSEnabled": false,
            "EncryptionEnabled": null,
            "Iops": null,
            "VolumeSize": null,
            "VolumeType": null
        },
        "ElasticsearchClusterConfig": {
            "DedicatedMasterCount": 3,
            "DedicatedMasterEnabled": true,
            "DedicatedMasterType": "m3.medium.elasticsearch",
            "InstanceCount": 6,
            "InstanceType": "m3.medium.elasticsearch",
            "ZoneAwarenessEnabled": true
        },
        "ElasticsearchVersion": "5.5",
        "EncryptionAtRestOptions": {
            "Enabled": true,
            "KmsKeyId": "arn:aws:kms:us-
west-1:123456789012:key/1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
        },
        "Endpoint": null,
        "Endpoints": null,
        "Processing": true,
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
        },
        "VPCOptions": {
            "AvailabilityZones": [
                "us-west-1b",
                "us-west-1c"
            ],
            "SecurityGroupIds": [
                "sg-12345678"
            ],
            "SubnetIds": [
                "subnet-87654321",
                "subnet-12345678"
            ],
            "VPCId": "vpc-12345678"
        }
    }
}
```

DeleteElasticsearchServiceRole

Deletes the service-linked role between Amazon ES and Amazon EC2. This role gives Amazon ES permissions to place VPC endpoints into your VPC. A service-linked role must be in place for domains with VPC endpoints to be created or function properly.

> **Note**
> This action only succeeds if no domains are using the service-linked role.

Syntax

```
DELETE /2015-01-01/es/role
```

Request Parameters

This operation does not use request parameters.

Request Body

This operation does not use the HTTP request body.

Response Elements

Not applicable. The `DeleteElasticsearchServiceRole` operation does not return a data structure.

Errors

`DeleteElasticsearchServiceRole` can return any of the following errors:

- `BaseException` (p.   )
- `InternalException` (p.   )
- `ValidationException` (p.   )

Example

The following example demonstrates deletion of the service-linked role:

Request

```
DELETE es.<AWS_REGION>.amazonaws.com/2015-01-01/es/role
```

Response

If successful, this action provides no response.

DescribeElasticsearchDomain

Describes the domain configuration for the specified Amazon ES domain, including the domain ID, domain service endpoint, and domain ARN.

Syntax

```
GET /2015-01-01/es/domain/<DOMAIN_NAME>
```

Request Parameters

Parameter	Data Type	Required?	Description
DomainName	DomainName (p. 213)	Yes	Name of the Amazon ES domain that you want to describe.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
DomainStatus	ElasticsearchDomainStatus (p. 215)	Configuration of the specified Amazon ES domain.

Errors

DescribeElasticsearchDomain can return any of the following errors:

- BaseException (p.)
- InternalException (p.)
- ResourceNotFoundException (p.)
- ValidationException (p.)

Example

The following example returns a description of the streaming-logs domain:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/domain/streaming-logs
```

Response

```
{
    "DomainStatus": {
        "ARN": "arn:aws:es:us-west-1:123456789012:domain/streaming-logs",
        "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-
west-1:123456789012:domain/streaming-logs/*\",\"Condition\":{\"IpAddress\":{\"aws:SourceIp
\":[\"11.222.333.11\",\"11.222.333.12\",\"11.222.333.13\",\"11.222.333.14\",
\"11.222.333.15\"]}}}]}",
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "Created": true,
        "Deleted": false,
        "DomainId": "123456789012/streaming-logs",
        "DomainName": "streaming-logs",
        "EBSOptions": {
            "EBSEnabled": true,
            "EncryptionEnabled": false,
            "Iops": null,
            "VolumeSize": 11,
            "VolumeType": "gp2"
        },
        "ElasticsearchClusterConfig": {
            "DedicatedMasterCount": 2,
            "DedicatedMasterEnabled": false,
            "DedicatedMasterType": "m4.large.elasticsearch",
            "InstanceCount": 2,
            "InstanceType": "t2.small.elasticsearch",
            "ZoneAwarenessEnabled": false
        },
        "ElasticsearchVersion": "5.5",
        "EncryptionAtRestOptions": {
            "Enabled": true,
            "KmsKeyId": "arn:aws:kms:us-
west-1:123456789012:key/1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
        },
        "CognitoOptions": {
            "IdentityPoolId": "us-west-1:12345678-1234-1234-1234-123456789012",
            "RoleArn": "arn:aws:iam::123456789012:role/my-kibana-role",
            "Enabled": true,
            "UserPoolId": "us-west-1_121234567"
        },
        "Endpoint": "search-streaming-logs-oojmrbhufr27n44zdri52wukdy.us-
west-1.es.amazonaws.com",
        "Endpoints": null,
        "Processing": false,
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 8
        },
        "VPCOptions": null
    }
}
```

DescribeElasticsearchDomainConfig

Displays the configuration of an Amazon ES domain.

Syntax

```
GET /2015-01-01/es/domain/<DOMAIN_NAME>/config
```

Request Parameters

Parameter	Data Type	Required?	Description
DomainName	DomainName (p. 213)	Yes	Name of the Amazon ES domain.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
DomainConfig	ElasticsearchDomainConfig (p. 215)	Configuration of the Amazon ES domain.

Errors

The DescribeElasticsearchDomainConfig operation can return any of the following errors:

- BaseException (p.)
- InternalException (p.)
- ResourceNotFoundException (p.)

Example

The following example returns a description of the configuration of the logs domain:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/domain/logs/config
```

Response

```
HTTP/1.1 200 OK
{
    "DomainConfig": {
        "AccessPolicies": {
            "Options": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":\"es:*\",\"Resource
\":\"arn:aws:es:us-west-1:123456789012:domain/logs/*\"}]}",
            "Status": {
                "CreationDate": 1499817484.04,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1500308955.652,
                "UpdateVersion": 17
            }
        },
        "AdvancedOptions": {
            "Options": {
                "indices.fielddata.cache.size": "",
                "rest.action.multi.allow_explicit_index": "true"
            },
            "Status": {
                "CreationDate": 1499817484.04,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1499818054.108,
                "UpdateVersion": 5
            }
        },
        "EBSOptions": {
            "Options": {
                "EBSEnabled": true,
                "EncryptionEnabled": false,
                "Iops": 0,
                "VolumeSize": 10,
                "VolumeType": "gp2"
            },
            "Status": {
                "CreationDate": 1499817484.04,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1499818054.108,
                "UpdateVersion": 5
            }
        },
        "ElasticsearchClusterConfig": {
            "Options": {
                "DedicatedMasterCount": 2,
                "DedicatedMasterEnabled": false,
                "DedicatedMasterType": "m4.large.elasticsearch",
                "InstanceCount": 2,
                "InstanceType": "m4.large.elasticsearch",
                "ZoneAwarenessEnabled": false
            },
            "Status": {
                "CreationDate": 1499817484.04,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1499966854.612,
                "UpdateVersion": 13
            }
        },
        "ElasticsearchVersion": {
            "Options": "5.5",
            "Status": {
                "PendingDeletion": false,
                "State": "Active",
                "CreationDate": 1436913638.995,
                "UpdateVersion": 6,
                "UpdateDate": 1436914324.278
            },
            "Options": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",
\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/logs/*\"}]}"
        },
        "EncryptionAtRestOptions": {
            "Options": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
west-1:123456789012:key/1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
            },
            "Status": {
                "CreationDate": 1509490412.757,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1509490953.717,
                "UpdateVersion": 6
            }
        },
        "LogPublishingOptions":{
            "Status":{
                "CreationDate":1502774634.546,
                "PendingDeletion":false,
                "State":"Processing",
                "UpdateDate":1502779590.448,
                "UpdateVersion":60
            },
            "Options":{
                "INDEX_SLOW_LOGS":{
                    "CloudWatchLogsLogGroupArn":"arn:aws:logs:us-east-1:123456789012:log-
group:sample-domain",
                    "Enabled":true
                },
                "SEARCH_SLOW_LOGS":{
                    "CloudWatchLogsLogGroupArn":"arn:aws:logs:us-east-1:123456789012:log-
group:sample-domain",
                    "Enabled":true
                }
            }
        },
        "SnapshotOptions": {
            "Options": {
                "AutomatedSnapshotStartHour": 6
            },
            "Status": {
                "CreationDate": 1499817484.04,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1499818054.108,
                "UpdateVersion": 5
            }
        },
        "VPCOptions": {
            "Options": {
                "AvailabilityZones": [
                    "us-west-1b"
                ],
                "SecurityGroupIds": [
                    "sg-12345678"
                ],
                "SubnetIds": [
                    "subnet-12345678"
                ],
                "VPCId": "vpc-12345678"
            },
            "Status": {
                "CreationDate": 1499817484.04,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1499818054.108,
                "UpdateVersion": 5
            }
        }
    }
}
```

DescribeElasticsearchDomains

Describes the domain configuration for up to five specified Amazon ES domains. Information includes the domain ID, domain service endpoint, and domain ARN.

Syntax

```
POST /2015-01-01/es/domain-info
{
    "DomainNames": [
        "<DOMAIN_NAME>",
        "<DOMAIN_NAME>",
    ]
}
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Field	Data Type	Required	Description
DomainNames	DomainNameList (p. 213)	Yes	Array of Amazon ES domains in the following format: {"DomainNames": ["<Domain_Name>","<Domain_Name>"...]

Response Elements

Field	Data Type	Description
DomainStatusList	ElasticsearchDomainStatusList (p. 217)	List that contains the status of each requested Amazon ES domain.

Errors

The `DescribeElasticsearchDomains` operation can return any of the following errors:

- `BaseException` (p.)
- `InternalException` (p.)
- `ValidationException` (p.)

Example

The following example returns a description of the `logs` and `streaming-logs` domains:

Request

```
POST es.<AWS_REGION>.amazonaws.com/2015-01-01/es/domain-info/
{
    "DomainNames": [
        "logs",
        "streaming-logs"
    ]
}
```

Response

```
HTTP/1.1 200 OK
{
    "DomainStatusList": [
        {
            "ElasticsearchClusterConfig": {
                "DedicatedMasterEnabled": true,
                "InstanceCount": 3,
                "ZoneAwarenessEnabled": false,
                "DedicatedMasterType": "m3.medium.elasticsearch",
                "InstanceType": "m3.medium.elasticsearch",
                "DedicatedMasterCount": 3
            },
            "ElasticsearchVersion": "5.5",
            "EncryptionAtRestOptions": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
west-1:123456789012:key/1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
            },
            "Endpoint": "search-streaming-logs-okga24ftzsbz2a2hzhsqw73jpy.us-
east-1.es.example.com",
            "Created": true,
            "Deleted": false,
            "DomainName": "streaming-logs",
            "EBSOptions": {
                "EBSEnabled": false
            },
            "VPCOptions": {
                "SubnetIds": [
                    "subnet-d1234567"
                ],
                "VPCId": "vpc-12345678",
                "SecurityGroupIds": [
                    "sg-123456789"
                ],
                "AvailabilityZones": [
                    "us-east-1"
                ]
            },
            "SnapshotOptions": {
                "AutomatedSnapshotStartHour": 0
            },
            "DomainId": "123456789012/streaming-logs",
            "AccessPolicies": "",
            "Processing": false,
            "AdvancedOptions": {
                "rest.action.multi.allow_explicit_index": "true",
                "indices.fielddata.cache.size": ""
            },
            "ARN": "arn:aws:es:us-east-1:123456789012:domain/streaming-logs"
        },
        {
            "ElasticsearchClusterConfig": {
                "DedicatedMasterEnabled": true,
                "InstanceCount": 1,
                "ZoneAwarenessEnabled": false,
                "DedicatedMasterType": "search.m3.medium",
                "InstanceType": "search.m3.xlarge",
                "DedicatedMasterCount": 3
            },
            "ElasticsearchVersion": "5.5",
            "EncryptionAtRestOptions": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
west-1:123456789012:key/1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
            },
            "Endpoint": "search-logs-p5st2kbt77diuihoqi6omd7jiu.us-east-1.es.example.com",
            "Created": true,
            "Deleted": false,
            "DomainName": "logs",
            "EBSOptions": {
                "Iops": 4000,
                "VolumeSize": 512,
                "VolumeType": "io1",
                "EBSEnabled": true
            },
            "VPCOptions": {
                "SubnetIds": [
                    "subnet-d1234567"
                ],
                "VPCId": "vpc-12345678",
                "SecurityGroupIds": [
                    "sg-123456789"
                ],
                "AvailabilityZones": [
                    "us-east-1"
                ]
            },
            "SnapshotOptions": {
                "AutomatedSnapshotStartHour": 0
            },
            "DomainId": "123456789012/logs",
            "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"\",
\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/logs/*\"}]}",
            "Processing": false,
            "AdvancedOptions": {
                "rest.action.multi.allow_explicit_index": "true"
            },
            "ARN": "arn:aws:es:us-east-1:123456789012:domain/logs"
        }
    ]
}
```

DescribeElasticsearchInstanceTypeLimits

Describes the instance count, storage, and master node limits for a given Elasticsearch version and instance type.

Syntax

```
GET 2015-01-01/es/instanceTypeLimits/{ElasticsearchVersion}/{InstanceType}?
domainName={DomainName}
```

Request Parameters

Parameter	Data Type	Required?	Description
ElasticsearchVersion	String	Yes	Elasticsearch version. For a list of supported versions, see the section called “Supported Elasticsearch Versions” (p. 2).
InstanceType	String	Yes	Instance type. To view instance types by region, see Amazon Elasticsearch Service Pricing.

Parameter	Data Type	Required?	Description
DomainName	DomainName (p. 213)	No	The name of an existing domain. Only specify if you need the limits for an existing domain.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
LimitsByRole	Map	Map containing all applicable instance type limits. "data" refers to data nodes. "master" refers to dedicated master nodes.

Errors

The DescribeElasticsearchInstanceTypeLimits operation can return any of the following errors:

- BaseException (p.)
- InternalException (p.)
- InvalidTypeException (p.)
- LimitExceededException (p.)
- ResourceNotFoundException (p.)
- ValidationException (p.)

Example

The following example returns a description of the logs and streaming-logs domains:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/instanceTypeLimits/6.0/
m4.large.elasticsearch
```

Response

```
HTTP/1.1 200 OK
{
    "LimitsByRole": {
        "data": {
            "AdditionalLimits": [
                {
                    "LimitName": "MaximumNumberOfDataNodesWithoutMasterNode",
                    "LimitValues": [
                        "10"
                    ]
                }
            ],
            "InstanceLimits": {
                "InstanceCountLimits": {
                    "MaximumInstanceCount": 20,
                    "MinimumInstanceCount": 1
                }
            },
            "StorageTypes": [
                {
                    "StorageSubTypeName": "standard",
                    "StorageTypeLimits": [
                        {
                            "LimitName": "MaximumVolumeSize",
                            "LimitValues": [
                                "100"
                            ]
                        },
                        {
                            "LimitName": "MinimumVolumeSize",
                            "LimitValues": [
                                "10"
                            ]
                        }
                    ],
                    "StorageTypeName": "ebs"
                },
                {
                    "StorageSubTypeName": "io1",
                    "StorageTypeLimits": [
                        {
                            "LimitName": "MaximumVolumeSize",
                            "LimitValues": [
                                "512"
                            ]
                        },
                        {
                            "LimitName": "MinimumVolumeSize",
                            "LimitValues": [
                                "35"
                            ]
                        },
                        {
                            "LimitName": "MaximumIops",
                            "LimitValues": [
                                "16000"
                            ]
                        },
                        {
                            "LimitName": "MinimumIops",
                            "LimitValues": [
                                "1000"
                            ]
                        }
                    ],
                    "StorageTypeName": "ebs"
                },
                {
                    "StorageSubTypeName": "gp2",
                    "StorageTypeLimits": [
                        {
                            "LimitName": "MaximumVolumeSize",
                            "LimitValues": [
                                "512"
                            ]
                        },
                        {
                            "LimitName": "MinimumVolumeSize",
                            "LimitValues": [
                                "10"
                            ]
                        }
                    ],
                    "StorageTypeName": "ebs"
                }
            ]
        },
        "master": {
            "AdditionalLimits": [
                {
                    "LimitName": "MaximumNumberOfDataNodesSupported",
                    "LimitValues": [
                        "100"
                    ]
                }
            ],
            "InstanceLimits": {
                "InstanceCountLimits": {
                    "MaximumInstanceCount": 5,
                    "MinimumInstanceCount": 2
                }
            },
            "StorageTypes": null
        }
    }
}
```

DescribeReservedElasticsearchInstanceOfferings

Describes the available Reserved Instance offerings for a given region.

Syntax

```
GET /2015-01-01/es/reservedInstanceOfferings?
offeringId={OfferingId}&maxResults={MaxResults}&nextToken={NextToken}
```

Request Parameters

Parameter	Data Type	Required?	Description
`OfferingId`	`String`	No	The offering ID.
`MaxResults`	`Integer`	No	Limits the number of results. Must be between 30 and 100.
`NextToken`	`String`	No	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
`ReservedElasticsearchInstanceOfferings`	`ReservedElasticsearchInstanceOfferings`	Container for all information on a Reserved Instance offering. To learn more, see the section called "Purchasing Reserved Instances (AWS CLI)" (p. 230).

Errors

The `DescribeReservedElasticsearchInstanceOfferings` operation can return any of the following errors:

- `DisabledOperationException` (p.)
- `InternalException` (p.)
- `ResourceNotFoundException` (p.)
- `ValidationException` (p.)

Example

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/reservedInstanceOfferings
```

Response

```
{
  "ReservedElasticsearchInstanceOfferings": [
    {
      "FixedPrice": 100.0,
      "ReservedElasticsearchInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.603,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "UsagePrice": 0.0,
      "PaymentOption": "PARTIAL_UPFRONT",
      "Duration": 31536000,
      "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",
      "CurrencyCode": "USD"
    }
  ]
}
```

DescribeReservedElasticsearchInstances

Describes the instances you have reserved in a given region.

Syntax

```
GET 2015-01-01/es/reservedInstances?
reservationId={ReservationId}&maxResults={PageSize}&nextToken={NextToken}
```

Request Parameters

Parameter	Data Type	Required?	Description
`ReservationId`	`String`	No	The reservation ID, assigned after you purchase a reservation.
`MaxResults`	`Integer`	No	Limits the number of results. Must be between 30 and 100.
`NextToken`	`String`	No	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
`ReservedElasticsearchInstances`	`ReservedElasticsearchInstances`	Container for all information on the instance you have reserved. To learn more, see the section called "Purchasing Reserved Instances (AWS CLI)" (p. 230).

Errors

The `DescribeReservedElasticsearchInstances` operation can return any of the following errors:

- `DisabledOperationException` (p.)
- `InternalException` (p.)
- `ResourceNotFoundException` (p.)
- `ValidationException` (p.)

Example

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/reservedInstances
```

Response

```
{
  "ReservedElasticsearchInstances": [
    {
      "FixedPrice": 100.0,
      "ReservedElasticsearchInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "ReservationName": "my-reservation",
      "PaymentOption": "PARTIAL_UPFRONT",
      "UsagePrice": 0.0,
      "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": 0.603,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "State": "payment-pending",
      "StartTime": 1522872571.229,
      "ElasticsearchInstanceCount": 3,
      "Duration": 31536000,
      "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",
      "CurrencyCode": "USD"
    }
  ]
}
```

GetCompatibleElasticsearchVersions

Returns a map of Elasticsearch versions and the versions you can upgrade them to.

Syntax

```
GET /2015-01-01/es/compatibleVersions?domainName={DomainName}
```

Request Parameters

Parameter	Data Type	Required?	Description
DomainName	DomainName (p. 213)	No	The name of an existing domain.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
ElasticsearchVersions	Map	A map of Elasticsearch versions and the versions you can upgrade them to.

Errors

The `GetCompatibleElasticsearchVersions` operation can return any of the following errors:

- BaseException (p.)
- ResourceNotFoundException (p.)
- DisabledOperationException (p.)
- ValidationException (p.)
- InternalException (p.)

Example

The following example lists all three domains owned by the current user:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/compatibleVersions
```

Response

```
{
    "CompatibleElasticsearchVersions": [
        {
            "SourceVersion": "6.0",
            "TargetVersions": [
                "6.3"
            ]
        },
        {
            "SourceVersion": "5.1",
            "TargetVersions": [
                "5.6"
            ]
        },
        {
            "SourceVersion": "6.2",
            "TargetVersions": [
                "6.3"
            ]
        },
        {
            "SourceVersion": "5.3",
            "TargetVersions": [
                "5.6"
            ]
        },
        {
            "SourceVersion": "5.5",
            "TargetVersions": [
                "5.6"
            ]
        },
        {
            "SourceVersion": "5.6",
            "TargetVersions": [
                "6.3"
            ]
        }
    ]
}
```

GetUpgradeHistory

Returns a list of the domain's 10 most-recent upgrade operations.

Syntax

```
GET /2015-01-01/es/upgradeDomain/{DomainName}/history?
maxResults={MaxResults}&amp;nextToken={NextToken}
```

Request Parameters

Parameter	Data Type	Required?	Description
`MaxResults`	`Integer`	No	Limits the number of results. Must be between 30 and 100.
`NextToken`	`String`	No	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
`UpgradeHistoryList`	`UpgradeHistoryList`	Container for result logs of the past ten upgrade operations.

Errors

The `GetCompatibleElasticsearchVersions` operation can return any of the following errors:

- BaseException (p.)
- ResourceNotFoundException (p.)
- DisabledOperationException (p.)
- ValidationException (p.)

- InternalException (p.   )

Example

The following example lists the upgrade history for the given domain:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/upgradeDomain/my-domain/history
```

Response

```
{
  "NextToken": null,
  "UpgradeHistories": [
    {
      "StartTimestamp": 1532466876,
      "StepsList": [
        {
          "Issues": [
            "Upgrade automated snapshot 00010e1cbc.2018-07-24t21-14-40 in state FAILED
 could not be completed successfully"
          ],
          "ProgressPercent": null,
          "UpgradeStep": "SNAPSHOT",
          "UpgradeStepStatus": "FAILED"
        },
        {
          "Issues": null,
          "ProgressPercent": null,
          "UpgradeStep": "PRE_UPGRADE_CHECK",
          "UpgradeStepStatus": "SUCCEEDED"
        }
      ],
      "UpgradeName": "Upgrade from 5.6 to 6.3",
      "UpgradeStatus": "FAILED"
    },
    {
      "StartTimestamp": 1532388708,
      "StepsList": [
        {
          "Issues": null,
          "ProgressPercent": null,
          "UpgradeStep": "PRE_UPGRADE_CHECK",
          "UpgradeStepStatus": "SUCCEEDED"
        }
      ],
      "UpgradeName": "Pre-Upgrade Check from 5.6 to 6.3",
      "UpgradeStatus": "SUCCEEDED"
    },
    {
      "StartTimestamp": 1532378327,
      "StepsList": [
        {
          "Issues": null,
          "ProgressPercent": null,
          "UpgradeStep": "UPGRADE",
          "UpgradeStepStatus": "SUCCEEDED"
        },
        {
          "Issues": null,
          "ProgressPercent": null,
          "UpgradeStep": "SNAPSHOT",
          "UpgradeStepStatus": "SUCCEEDED"
        },
        {
          "Issues": null,
          "ProgressPercent": null,
          "UpgradeStep": "PRE_UPGRADE_CHECK",
          "UpgradeStepStatus": "SUCCEEDED"
        }
      ],
      "UpgradeName": "Upgrade from 5.3 to 5.6",
      "UpgradeStatus": "SUCCEEDED"
    }
  ]
}
```

GetUpgradeStatus

Returns the most-recent status of a domain's Elasticsearch version upgrade.

Syntax

```
GET /2015-01-01/es/upgradeDomain/{DomainName}/status
```

Request Parameters

Parameter	Data Type	Required?	Description
`DomainName`	`DomainName` (p. 213)	Yes	The name of an existing domain.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
`UpgradeStepItem`	`UpgradeStepItem`	Container for the most-recent status of a domain's version upgrade.

Errors

The `GetCompatibleElasticsearchVersions` operation can return any of the following errors:

- BaseException (p.)
- ResourceNotFoundException (p.)
- DisabledOperationException (p.)

- ValidationException (p.)
- InternalException (p.)

Example

The following example lists the upgrade status for the given domain:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/upgradeDomain/my-domain/status
```

Response

```
{
  "StepStatus": "FAILED",
  "UpgradeName": "Upgrade from 5.6 to 6.3",
  "UpgradeStep": "SNAPSHOT"
}
```

ListDomainNames

Displays the names of all Amazon ES domains owned by the current user *in the active region*.

Syntax

```
GET /2015-01-01/domain
```

Request Parameters

This operation does not use request parameters.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	**Data Type**	**Description**
`DomainNameList`	`DomainNameList` (p. 213)	The names of all Amazon ES domains owned by the current user.

Errors

The `ListDomainNames` operation can return any of the following errors:

- BaseException (p.)

- ValidationException (p.)

Example

The following example lists all three domains owned by the current user:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/domain
```

Response

```
{
    "DomainNames": [
        {
            "DomainName": "logs"
        },
        {
            "DomainName": "streaming-logs"
        }
    ]
}
```

ListElasticsearchInstanceTypeDetails

Lists all Elasticsearch instance types that are supported for a given Elasticsearch version and the features that these instance types support.

Syntax

```
GET 2015-01-01/es/instanceTypeDetails/{ElasticsearchVersion}?
domainName={DomainName}&maxResults={MaxResults}&nextToken={NextToken}
```

Request Parameters

Parameter	Data Type	Required?	Description
ElasticsearchVersion	String	Yes	The Elasticsearch version.
DomainName	String	No	The Amazon ES domain name.
MaxResults	Integer	No	Limits the number of results. Must be between 30 and 100.
NextToken	String	No	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
`ElasticsearchInstanceTypes`	`List`	List of supported instance types for the given Elasticsearch version and the features that these instance types support.
`NextToken`	`String`	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Errors

`ListElasticsearchInstanceTypeDetails` can return any of the following errors:

- `BaseException` (p.)
- `InternalException` (p.)
- `ResourceNotFoundException` (p.)
- `ValidationException` (p.)

Example

Request

```
GET es.us-west-1.amazonaws.com/2015-01-01/es/instanceTypeDetails/6.2
```

Response

```
{
    "ElasticsearchInstanceTypeDetails": [
        {
            "AppLogsEnabled": true,
            "CognitoEnabled": true,
            "EncryptionEnabled": false,
            "InstanceType": "t2.small.elasticsearch"
        },
        {
            "AppLogsEnabled": true,
            "CognitoEnabled": true,
            "EncryptionEnabled": false,
            "InstanceType": "t2.medium.elasticsearch"
        },
        {
            "AppLogsEnabled": true,
            "CognitoEnabled": true,
            "EncryptionEnabled": true,
            "InstanceType": "c4.large.elasticsearch"
        },
        {
            "AppLogsEnabled": true,
            "CognitoEnabled": true,
            "EncryptionEnabled": true,
            "InstanceType": "c4.xlarge.elasticsearch"
        },
        ...
    ],
    "NextToken": null
}
```

ListElasticsearchInstanceTypes (Deprecated)

Lists all Elasticsearch instance types that are supported for a given Elasticsearch version. This action is deprecated. Use ListElasticsearchInstanceTypeDetails (p. 196) instead.

Syntax

```
GET 2015-01-01/es/instanceTypes/{ElasticsearchVersion}?
domainName={DomainName}&maxResults={MaxResults}&nextToken={NextToken}
```

Request Parameters

Parameter	Data Type	Required?	Description
ElasticsearchVersion	String	Yes	The Elasticsearch version.
DomainName	String	No	The Amazon ES domain name.
MaxResults	Integer	No	Limits the number of results. Must be between 30 and 100.
NextToken	String	No	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
ElasticsearchInstanceTypes	List	List of supported instance types for the given Elasticsearch version.

Field	Data Type	Description
`NextToken`	`String`	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Errors

`ListElasticsearchInstanceTypes` can return any of the following errors:

- `BaseException` (p.)
- `InternalException` (p.)
- `ResourceNotFoundException` (p.)
- `ValidationException` (p.)

Example

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/instanceTypes/6.0
```

Response

```
{
    "ElasticsearchInstanceTypes": [
        "t2.small.elasticsearch",
        "t2.medium.elasticsearch",
        "r4.large.elasticsearch",
        "r4.xlarge.elasticsearch",
        "r4.2xlarge.elasticsearch",
        "r4.4xlarge.elasticsearch",
        "r4.8xlarge.elasticsearch",
        "r4.16xlarge.elasticsearch",
        "m4.large.elasticsearch",
        "m4.xlarge.elasticsearch",
        "m4.2xlarge.elasticsearch",
        "m4.4xlarge.elasticsearch",
        "m4.10xlarge.elasticsearch",
        "c4.large.elasticsearch",
        "c4.xlarge.elasticsearch",
        "c4.2xlarge.elasticsearch",
        "c4.4xlarge.elasticsearch",
        "c4.8xlarge.elasticsearch"
    ],
    "NextToken": null
}
```

ListElasticsearchVersions

Lists all supported Elasticsearch versions on Amazon ES.

Syntax

```
GET 2015-01-01/es/versions?maxResults={MaxResults}&nextToken={NextToken}
```

Request Parameters

Parameter	Data Type	Required?	Description
MaxResults	Integer	No	Limits the number of results. Must be between 30 and 100.
NextToken	String	No	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
ElasticsearchVersions	List	Lists all supported Elasticsearch versions.
NextToken	String	Used for pagination. Only necessary if a previous API call produced a result containing NextToken. Accepts a next-token input to return results for the next page and provides a next-token output in the response, which clients can use to retrieve more results.

Errors

ListElasticsearchVersions can return any of the following errors:

- BaseException (p.)
- InternalException (p.)
- ResourceNotFoundException (p.)
- ValidationException (p.)

Example

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/es/versions
```

Response

```
{
    "ElasticsearchVersions": [
        "6.0",
        "5.5",
        "5.3",
        "5.1",
        "2.3",
        "1.5"
    ],
    "NextToken": null
}
```

ListTags

Displays all resource tags for an Amazon ES domain.

Syntax

```
GET /2015-01-01/tags?arn=<DOMAIN_ARN>
```

Request Parameters

Parameter	Data Type	Required?	Description
ARN	ARN (p. 211)	Yes	Amazon Resource Name (ARN) for the Amazon ES domain.

Request Body

This operation does not use the HTTP request body.

Response Elements

Field	Data Type	Description
TagList	TagList (p. 222)	List of resource tags. For more information, see Tagging Amazon Elasticsearch Service Domains (p. 55).

Errors

The ListTags operation can return any of the following errors:

- BaseException (p.)
- ResourceNotFoundException (p.)
- ValidationException (p.)
- InternalException (p.)

Example

The following example lists the tags attached to the `logs` domain:

Request

```
GET es.<AWS_REGION>.amazonaws.com/2015-01-01/tags?arn=arn:aws:es:us-
west-1:123456789012:domain/logs
```

Response

```
HTTP/1.1 200 OK
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "MacOS"
        },
        {
            "Key": "project",
            "Value": "trident"
        }
    ]
}
```

PurchaseReservedElasticsearchInstance

Purchases a Reserved Instance.

Syntax

```
POST /2015-01-01/es/purchaseReservedInstanceOffering
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Name	Data Type	Required?	Description
`ReservationName`	String	Yes	A descriptive name for your reservation.
`ReservedElasticsearchInstanceOfferingId`	String	Yes	The offering ID.
`InstanceCount`	Integer	Yes	The number of instances you want to reserve.

Response Elements

Field	Data Type	Description
`ReservationName`	String	The name of your reservation.

Field	Data Type	Description
`ReservedElasticsearchInstanceId`	String	The reservation ID.

Errors

The `PurchaseReservedElasticsearchInstance` operation can return any of the following errors:

- `DisabledOperationException` (p.)
- `InternalException` (p.)
- `ResourceNotFoundException` (p.)
- `ValidationException` (p.)
- `LimitExceededException` (p.)
- `ResourceAlreadyExistsException` (p.)

Example

Request

```
POST es.<AWS_REGION>.amazonaws.com/2015-01-01/es/purchaseReservedInstanceOffering
{
  "ReservationName" : "my-reservation",
  "ReservedElasticsearchInstanceOfferingId" : "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
  "InstanceCount" : 3
}
```

Response

```
{
  "ReservationName": "my-reservation",
  "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a"
}
```

RemoveTags

Removes the specified resource tags from an Amazon ES domain.

Syntax

```
POST es.<AWS_REGION>.amazonaws.com/2015-01-01/tags-removal
{
    "ARN": "<DOMAIN_ARN>",
    "TagKeys": [
        "<TAG_KEY>",
        "<TAG_KEY>",
        ...
    ]
}
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Parameter	Data Type	Required?	Description
ARN	ARN (p. 211)	Yes	Amazon Resource Name (ARN) of an Amazon ES domain. For more information, see Identifiers for IAM Entities in *Using AWS Identity and Access Management*.
TagKeys	TagKey (p. 221)	Yes	List of tag keys for resource tags that you want to remove from an Amazon ES domain.

Response Elements

Not applicable. The RemoveTags operation does not return a response element.

Errors

The RemoveTags operation can return any of the following errors:

- BaseException (p.)
- ValidationException (p.)
- InternalException (p.)

Example

The following example deletes a resource tag with a tag key of project from the Amazon ES domain:

Request

```
POST /2015-01-01/tags-removal
{
    "ARN": "<DOMAIN_ARN>",
    "TagKeys": [
        "project"
    ]
}
```

This operation does not return a response element.

UpdateElasticsearchDomainConfig

Modifies the configuration of an Amazon ES domain, such as the instance type and the number of instances. You only need to specify the values that you want to update.

Syntax

```
POST /2015-01-01/es/domain/<DOMAIN_NAME>/config
{
    "ElasticsearchClusterConfig": {
        "InstanceType": "<INSTANCE_TYPE>",
        "Instance_Count": <INSTANCE_COUNT>,
        "DedicatedMasterEnabled": "<TRUE|FALSE>",
        "DedicatedMasterCount": <INSTANCE_COUNT>,
        "DedicatedMasterType": "<INSTANCE_COUNT>",
        "ZoneAwarenessEnabled": "<TRUE|FALSE>"
    },
    "EBSOptions": {
        "EBSEnabled": "<TRUE|FALSE>",
        "VolumeType": "<VOLUME_TYPE>",
        "VolumeSize": "<VOLUME_SIZE>",
        "Iops": "<VALUE>"
    },
    "VPCOptions": {
        "SubnetIds": [
            "<SUBNET_ID>"
        ],
        "SecurityGroupIds": [
            "<SECURITY_GROUP_ID>"
        ]
    },
    "AccessPolicies": "<ACCESS_POLICY_DOCUMENT>",
    "SnapshotOptions": {
        "AutomatedSnapshotStartHour": <START_HOUR>,
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "<TRUE|FALSE>",
            "indices.fielddata.cache.size": "<PERCENTAGE_OF_HEAP>"
        }
    },
    "LogPublishingOptions": {
        "SEARCH_SLOW_LOGS": {
            "CloudWatchLogsLogGroupArn":"<ARN>",
            "Enabled":true
        },
        "INDEX_SLOW_LOGS": {
            "CloudWatchLogsLogGroupArn":"<ARN>",
            "Enabled":true
        }
    }
}
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Parameter	Data Type	Required	Description
DomainName	DomainName (p. 213)	Yes	Name of the Amazon ES domain for which you want to update the configuration.
ElasticsearchClusterConfig	ElasticsearchClusterConfig (p. 214)	No	Changes that you want to make to the cluster configuration, such as the instance type and number of EC2 instances.
EBSOptions	EBSOptions (p. 214)	No	Type and size of EBS volumes attached to data nodes.
VPCOptions	VPCOptions (p. 223)	No	Container for the values required to configure Amazon ES to work with a VPC. To learn more, see VPC Support for Amazon Elasticsearch Service Domains (p. 117).

Parameter	Data Type	Required	Description
SnapshotOptions	SnapshotOptions (p. 221)	No	Hour during which the service takes an automated daily snapshot of the indices in the Amazon ES domain.
AdvancedOptions	AdvancedOptions (p. 210)	No	Key-value pairs to specify advanced configuration options. For more information, see Configuring Advanced Options (p. 24).
AccessPolicies	String	No	Specifies the access policies for the Amazon ES domain. For more information, see Configuring Access Policies (p. 21).
LogPublishingOptions	LogPublishingOptions (p. 218)	No	Key-value string pairs to configure slow log publishing.
CognitoOptions	CognitoOptions (p. 212)	No	Key-value pairs to configure Amazon ES to use Amazon Cognito authentication for Kibana.

Response Elements

Field	Data Type	Description
DomainConfig	String	Status of the Amazon ES domain after updating its configuration.

Errors

UpdateElasticsearchDomainConfig can return any of the following errors:

- BaseException (p.)
- InternalException (p.)
- InvalidTypeException (p.)
- LimitExceededException (p.)
- ValidationException (p.)

Example

The following example configures the daily automatic snapshot for the streaming-logs domain to occur during the hour starting at 3:00 AM GMT:

Request

```
POST es.<AWS_REGION>.amazonaws.com/2015-01-01/es/domain/streaming-logs/config
{
    "SnapshotOptions": {
        "AutomatedSnapshotStartHour": 3
    }
}
```

Response

```
{
    "DomainConfig": {
        "AccessPolicies": {
            "Options": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"*\"},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-
west-1:123456789012:domain/streaming-logs/*\",\"Condition\":{\"IpAddress\":{\"aws:SourceIp
\":[\"11.222.333.11\",\"11.222.333.12\",\"11.222.333.13\",\"11.222.333.14\",
\"11.222.333.15\"]}}}]}",
            "Status": {
                "CreationDate": 1502213150.329,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1502213466.93,
                "UpdateVersion": 6
            }
        },
        "AdvancedOptions": {
            "Options": {
                "rest.action.multi.allow_explicit_index": "true"
            },
            "Status": {
                "CreationDate": 1502213150.329,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1502213466.93,
                "UpdateVersion": 6
            }
        },
        "EBSOptions": {
            "Options": {
                "EBSEnabled": true,
                "EncryptionEnabled": false,
                "Iops": null,
                "VolumeSize": 11,
                "VolumeType": "gp2"
            },
            "Status": {
                "CreationDate": 1502213150.329,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1502929669.653,
                "UpdateVersion": 23
            }
        },
        "ElasticsearchClusterConfig": {
            "Options": {
                "DedicatedMasterCount": 2,
                "DedicatedMasterEnabled": false,
                "DedicatedMasterType": "m4.large.elasticsearch",
                "InstanceCount": 2,
                "InstanceType": "t2.small.elasticsearch",
                "ZoneAwarenessEnabled": false
            },
            "Status": {
                "CreationDate": 1502213150.329,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1502929669.653,
                "UpdateVersion": 23
            }
        },
        "ElasticsearchVersion": {
            "Options": "5.5",
            "Status": {
                "CreationDate": 1502213150.329,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1502213466.93,
                "UpdateVersion": 6
            }
        },
        "EncryptionAtRestOptions": {
            "Options": {
                "Enabled": true,
                "KmsKeyId": "arn:aws:kms:us-
west-1:123456789012:key/1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
            },
            "Status": {
                "CreationDate": 1509490412.757,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1509490953.717,
                "UpdateVersion": 6
            }
        },
        "LogPublishingOptions":{
            "Options":{
                "INDEX_SLOW_LOGS":{
                    "CloudWatchLogsLogGroupArn":"arn:aws:logs:us-east-1:123456789012:log-
group:sample-domain",
                    "Enabled":true
                },
                "SEARCH_SLOW_LOGS":{
                    "CloudWatchLogsLogGroupArn":"arn:aws:logs:us-east-1:123456789012:log-
group:sample-domain",
                    "Enabled":true
                }
            },
            "Status":{
                "CreationDate":1502774634.546,
                "PendingDeletion":false,
                "State":"Processing",
                "UpdateDate":1502779590.448,
                "UpdateVersion":60
            }
        },
        "SnapshotOptions": {
            "Options": {
                "AutomatedSnapshotStartHour": 3
            },
            "Status": {
                "CreationDate": 1502213150.329,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1503093165.447,
                "UpdateVersion": 25
            }
        },
        "VPCOptions": {
            "Options": {
                "AvailabilityZones": null,
                "SecurityGroupIds": null,
                "SubnetIds": null,
                "VPCId": null
            },
            "Status": {
                "CreationDate": 1503093165.597,
                "PendingDeletion": false,
                "State": "Active",
                "UpdateDate": 1503093165.597,
                "UpdateVersion": 25
            }
        }
    }
}
```

UpgradeElasticsearchDomain

Upgrades an Amazon ES domain to a new version of Elasticsearch. Alternately, checks upgrade eligibility.

Syntax

```
POST /2015-01-01/es/upgradeDomain
{
  "DomainName": "String",
  "TargetVersion": "String",
  "PerformCheckOnly": true|false
}
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Parameter	Data Type	Required	Description
`DomainName`	String	Yes	Name of the Amazon ES domain that you want to upgrade.
`TargetVersion`	String	Yes	Elasticsearch version to which you want to upgrade. See the section called “GetCompatibleElasticsearchVersions” (p. 190).
PerformCheckOnly	Boolean	No	Defaults to false. If true, Amazon ES checks the eligibility of the domain, but does not perform the upgrade.

Response Elements

Field	Data Type	Description
`UpgradeElasticsearchDomainResponse`	Map	Basic response confirming operation details.

Errors

`UpdateElasticsearchDomainConfig` can return any of the following errors:

- `BaseException` (p.)
- `ResourceNotFound` (p.)
- `ResourceAlreadyExists` (p.)
- `DisabledOperation` (p.)
- `ValidationException` (p.)
- `Internal` (p.)

Example

The following example upgrades an Amazon ES 5.*x* domain to Elasticsearch 5.6:

Request

```
POST es.<AWS_REGION>.amazonaws.com/2015-01-01/es/upgradeDomain/
{
  "DomainName": "my-domain",
  "TargetVersion": "5.6",
  "PerformCheckOnly": false
}
```

Response

```
{
  "DomainName": null,
  "PerformCheckOnly": null,
  "TargetVersion": null,
  "UpgradeId": null
}
```

Data Types

This section describes the data types used by the REST Configuration API.

AdvancedOptions

Key-value string pairs to specify advanced Elasticsearch configuration options.

Field	Data Type	Description
`rest.action.multi.allow_explicit_index`	Key-value pair: `rest.action.multi.allow_explicit_index=<true|false>`	Specifies whether explicit references to indices are allowed inside the body of HTTP requests. If you want to configure access policies for domain sub-resources, such as specific indices and domain APIs, you must disable this property. For more

Field	Data Type	Description
		information, see URL-based Access Control. For more information about access policies for sub-resources, see Configuring Access Policies (p. 21).
`indices.fielddata.cache.size`	Key-value pair: `indices.fielddata.cache.size=<percentage_of_heap>`	Specifies the percentage of Java heap space that is allocated to field data. By default, this setting is unbounded.
`indices.query.bool.max_clause_count`	Key-value pair: `indices.query.bool.max_clause_count=<int>`	Specifies the maximum number of clauses allowed in a Lucene Boolean query. 1024 is the default. Queries with more than the permitted number of clauses result in a `TooManyClauses` error. To learn more, see the Lucene documentation.

AdvancedOptionsStatus

Status of an update to advanced configuration options for an Amazon ES domain.

Field	Data Type	Description
Options	`AdvancedOptions` (p. 210)	Key-value pairs to specify advanced Elasticsearch configuration options.
Status	`OptionStatus` (p. 220)	Status of an update to advanced configuration options for an Amazon ES domain.

ARN

Field	Data Type	Description
`ARN`	String	Amazon Resource Name (ARN) of an Amazon ES domain. For more information, see IAM ARNs in the AWS Identity and Access Management documentation.

CognitoOptions

Field	Data Type	Description
`Enabled`	Boolean	Whether to enable or disable Amazon Cognito authentication for Kibana. See *Authentication for Kibana* (p. 104).
`UserPoolId`	String	The Amazon Cognito user pool ID that you want Amazon ES to use for Kibana authentication.
`IdentityPoolId`	String	The Amazon Cognito identity pool ID that you want Amazon ES to use for Kibana authentication.
`RoleArn`	String	The `AmazonESCognitoAccess` role that allows Amazon ES to configure your user pool and identity pool.

CognitoOptionsStatus

Field	Data Type	Description
`Options`	`CognitoOptions` (p. 212)	Key-value pairs to configure Amazon ES to use Amazon Cognito authentication for Kibana.
`Status`	`OptionStatus` (p. 220)	Status of an update to the Amazon Cognito configuration options for an Amazon ES domain.

CreateElasticsearchDomainRequest

Container for the parameters required by the `CreateElasticsearchDomain` service operation.

Field	Data Type	Description
`DomainName`	`DomainName` (p. 213)	Name of the Amazon ES domain to create.
`ElasticsearchClusterConfig`	`ElasticsearchClusterConfig` (p. 214)	Container for the cluster configuration of an Amazon ES domain.
`EBSOptions`	`EBSOptions` (p. 214)	Container for the parameters required to enable EBS-based storage for an Amazon ES domain. For more information, see Configuring EBS-based Storage (p. 18).
`AccessPolicies`	String	IAM policy document specifying the access policies for the new Amazon ES domain. For more information, see Configuring Access Policies (p. 21).
`SnapshotOptions`	`SnapshotOptionsStatus` (p. 221)	Container for parameters required to configure automated snapshots of

Field	Data Type	Description
		domain indices. For more information, see Configuring Snapshots (p. 23).
VPCOptions	VPCOptions (p. 223)	Container for the values required to configure Amazon ES to work with a VPC.
LogPublishingOptions	LogPublishingOptions (p. 218)	Key-value string pairs to configure slow log publishing.
SnapshotOptions	SnapshotOptionsStatus (p. 221)	Container for parameters required to configure automated snapshots of domain indices. For more information, see Configuring Snapshots (p. 23).
AdvancedOptions	AdvancedOptionsStatus (p. 211)	Key-value pairs to specify advanced configuration options.
CognitoOptions	CognitoOptions (p. 212)	Key-value pairs to configure Amazon ES to use Amazon Cognito authentication for Kibana.
NodeToNodeEncryptionOptions	NodeToNodeEncryptionOptions (p. 219)	Specify true to enable node-to-node encryption.

DomainID

Data Type	Description
String	Unique identifier for an Amazon ES domain

DomainName

Name of an Amazon ES domain.

Data Type	Description
String	Name of an Amazon ES domain. Domain names are unique across all domains owned by the same account within an AWS region. Domain names must start with a lowercase letter and must be between 3 and 28 characters. Valid characters are a-z (lowercase only), 0-9, and – (hyphen).

DomainNameList

String of Amazon ES domain names.

Data Type	Description
String Array	Array of Amazon ES domains in the following format: `["<Domain_Name>","<Domain_Name>"...]`

EBSOptions

Container for the parameters required to enable EBS-based storage for an Amazon ES domain. For more information, see Configuring EBS-based Storage (p. 18).

Field	Data Type	Description
`EBSEnabled`	Boolean	Indicates whether EBS volumes are attached to data nodes in an Amazon ES domain.
`VolumeType`	String	Specifies the type of EBS volumes attached to data nodes.
`VolumeSize`	String	Specifies the size of EBS volumes attached to data nodes.
`Iops`	String	Specifies the baseline input/output (I/O) performance of EBS volumes attached to data nodes. Applicable only for the Provisioned IOPS EBS volume type.

ElasticsearchClusterConfig

Container for the cluster configuration of an Amazon ES domain.

Field	Data Type	Description
`InstanceType`	String	Instance type of data nodes in the cluster.
`InstanceCount`	Integer	Number of instances in the cluster.
`DedicatedMasterEnabled`	Boolean	Indicates whether dedicated master nodes are enabled for the cluster. `True` if the cluster will use a dedicated master node. `False` if the cluster will not. For more information, see About Dedicated Master Nodes (p. 128).
`DedicatedMasterType`	String	Amazon ES instance type of the dedicated master nodes in the cluster.
`DedicatedMasterCount`	Integer	Number of dedicated master nodes in the cluster.
`ZoneAwarenessEnabled`	Boolean	Indicates whether zone awareness is enabled. Zone awareness allocates the nodes and replica index shards belonging to a cluster across two Availability Zones in the same region. If you enable zone awareness, you must have an even number of instances in the instance count, and you also must use the Amazon ES Configuration API to replicate your data for your Elasticsearch cluster. For more information, see Enabling Zone Awareness (p. 43).

ElasticsearchDomainConfig

Container for the configuration of an Amazon ES domain.

Field	Data Type	Description
ElasticsearchVersion	String	Elasticsearch version.
ElasticsearchClusterConfig	ElasticsearchClusterConfig (p. 214)	Container for the cluster configuration of an Amazon ES domain.
EBSOptions	EBSOptions (p. 214)	Container for EBS options configured for an Amazon ES domain.
AccessPolicies	String	Specifies the access policies for the Amazon ES domain. For more information, see Configuring Access Policies (p. 21).
SnapshotOptions	SnapshotOptionsStatus (p. 221)	Hour during which the service takes an automated daily snapshot of the indices in the Amazon ES domain. For more information, see Configuring Snapshots (p. 23).
VPCOptions	VPCDerivedInfoStatus (p. 222)	The current VPCOptions (p. 223) for the domain and the status of any updates to their configuration.
LogPublishingOptions	LogPublishingOptions (p. 218)	Key-value pairs to configure slow log publishing.
AdvancedOptions	AdvancedOptionsStatus (p. 210)	Key-value pairs to specify advanced configuration options.
EncryptionAtRestOptions	EncryptionAtRestOptionsStatus (p. 218)	Key-value pairs to enable encryption at rest.
NodeToNodeEncryptionOptions	NodeToNodeEncryptionOptionsStatus (p. 219)	Whether node-to-node encryption is enabled or disabled.

ElasticsearchDomainStatus

Container for the contents of a DomainStatus data structure.

Field	Data Type	Description
DomainID	DomainID (p. 213)	Unique identifier for an Amazon ES domain.

Field	Data Type	Description
`DomainName`	`DomainName` (p. 213)	Name of an Amazon ES domain. Domain names are unique across all domains owned by the same account within an AWS Region. Domain names must start with a lowercase letter and must be between 3 and 28 characters. Valid characters are a-z (lowercase only), 0-9, and – (hyphen).
`ARN`	`ARN` (p. 211)	Amazon Resource Name (ARN) of an Amazon ES domain. For more information, see Identifiers for IAM Entities in *Using AWS Identity and Access Management*.
`Created`	Boolean	Status of the creation of an Amazon ES domain. `True` if creation of the domain is complete. `False` if domain creation is still in progress.
`Deleted`	Boolean	Status of the deletion of an Amazon ES domain. `True` if deletion of the domain is complete. `False` if domain deletion is still in progress.
`Endpoint`	`ServiceUrl` (p. 220)	Domain-specific endpoint used to submit index, search, and data upload requests to an Amazon ES domain.
`Endpoints`	`EndpointsMap` (p. 218)	The key-value pair that exists if the Amazon ES domain uses VPC endpoints.
`Processing`	Boolean	Status of a change in the configuration of an Amazon ES domain. `True` if the service is still processing the configuration changes. `False` if the configuration change is active. You must wait for a domain to reach active status before submitting index, search, and data upload requests.
`ElasticsearchVersion`	String	Elasticsearch version.
`ElasticsearchClusterConfig`	`ElasticsearchClusterConfig` (p. 214)	Container for the cluster configuration of an Amazon ES domain.

Field	Data Type	Description
EBSOptions	EBSOptions (p. 214)	Container for the parameters required to enable EBS-based storage for an Amazon ES domain. For more information, see Configuring EBS-based Storage (p. 18).
AccessPolicies	String	IAM policy document specifying the access policies for the new Amazon ES domain. For more information, see Configuring Access Policies (p. 21).
SnapshotOptions	SnapshotOptions (p. 221)	Container for parameters required to configure the time of daily automated snapshots of Amazon ES domain indices.
VPCOptions	VPCDerivedInfo (p. 223)	Information that Amazon ES derives based on VPCOptions (p. 223) for the domain.
LogPublishingOptions	LogPublishingOptions (p. 218)	Key-value pairs to configure slow log publishing.
AdvancedOptions	AdvancedOptions (p. 210)	Key-value pairs to specify advanced configuration options.
EncryptionAtRestOptions	EncryptionAtRestOptions (p. 218)	Key-value pairs to enable encryption at rest.
CognitoOptions	CognitoOptions (p. 212)	Key-value pairs to configure Amazon ES to use Amazon Cognito authentication for Kibana.
NodeToNodeEncryptionOptions	NodeToNodeEncryptionOptions (p. 219)	Whether node-to-node encryption is enabled or disabled.

ElasticsearchDomainStatusList

List that contains the status of each specified Amazon ES domain.

Field	Data Type	Description
DomainStatusList	ElasticsearchDomainStatus (p. 215)	List that contains the status of each specified Amazon ES domain.

EncryptionAtRestOptions

Specifies whether the domain should encrypt data at rest, and if so, the AWS Key Management Service (KMS) key to use. Can only be used to create a new domain, not update an existing one.

Field	Data Type	Description
`Enabled`	Boolean	Specify `true` to enable encryption at rest.
`KmsKeyId`	String	The KMS key ID. Takes the form `1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6`

EncryptionAtRestOptionsStatus

Status of the domain's encryption at rest options.

Field	Data Type	Description
`Options`	`EncryptionAtRestOptions` (p. 218)	Encryption at rest options for the domain.
`Status`	`OptionStatus` (p. 220)	Status of the domain's encryption at rest options.

EndpointsMap

The key-value pair that contains the VPC endpoint. Only exists if the Amazon ES domain resides in a VPC.

Field	Data Type	Description
`Endpoints`	Key-value string pair: `"vpc": "<VPC_ENDPOINT>"`	The VPC endpoint for the domain.

LogPublishingOptions

Specifies whether the Amazon ES domain publishes the Elasticsearch application and slow logs to Amazon CloudWatch. You still have to enable the *collection* of slow logs using the Elasticsearch REST API. To learn more, see the section called “Setting Elasticsearch Logging Thresholds for Slow Logs” (p. 28).

Field	Data Type	Description
`INDEX_SLOW_LOGS`	Key-value	Two key-value pairs that define the CloudWatch log group and whether the Elasticsearch index slow log should be published there:

Field	Data Type	Description

		`"CloudWatchLogsLogGroupArn":"arn:aws:logs:us-`
		`east-1:264071961897:log-group:sample-`
		`domain",`
		`"Enabled":true`

`SEARCH_SLOW_LOGS`	Key-value	Two key-value pairs that define the CloudWatch log group and whether the Elasticsearch search slow log should be published there:
		`"CloudWatchLogsLogGroupArn":"arn:aws:logs:us-`
		`east-1:264071961897:log-group:sample-`
		`domain",`
		`"Enabled":true`

`ES_APPLICATION_LOGS`	Key-value	Two key-value pairs that define the CloudWatch log group and whether the Elasticsearch error logs should be published there:
		`"CloudWatchLogsLogGroupArn":"arn:aws:logs:us-`
		`east-1:264071961897:log-group:sample-`
		`domain",`
		`"Enabled":true`

LogPublishingOptionsStatus

Status of an update to the configuration of the slow log publishing options for the Amazon ES domain.

Field	Data Type	Description
`Options`	`LogPublishingOptions` (p. 218)	Log publishing options for the domain
`Status`	`OptionStatus` (p. 220)	Status of an update to snapshot options for an Amazon ES domain

NodeToNodeEncryptionOptions

Enables or disables node-to-node encryption.

Field	Data Type	Description
`Enabled`	Boolean	Enable with `true`.

NodeToNodeEncryptionOptionsStatus

State of a domain's node-to-node encryption options.

Field	Data Type	Description
Options	NodeToNodeEncryptionOptions (p. 219)	Whether node-to-node encryption is enabled or disabled.
Status	OptionStatus (p. 220)	Status of the setting.

OptionState

State of an update to advanced options for an Amazon ES domain.

Field	Data Type	Description
OptionStatus	String	One of three valid values: • RequiresIndexDocuments • Processing • Active

OptionStatus

Status of an update to configuration options for an Amazon ES domain.

Field	Data Type	Description
CreationDate	Time stamp	Date and time when the Amazon ES domain was created
UpdateDate	Time stamp	Date and time when the Amazon ES domain was updated
UpdateVersion	Integer	Whole number that specifies the latest version for the entity
State	OptionState (p. 220)	State of an update to configuration options for an Amazon ES domain
PendingDeletion	Boolean	Indicates whether the service is processing a request to permanently delete the Amazon ES domain and all of its resources

ServiceURL

Domain-specific endpoint used to submit index, search, and data upload requests to an Amazon ES domain.

Field	Data Type	Description
ServiceURL	String	Domain-specific endpoint used to submit index, search, and data upload requests to an Amazon ES domain

SnapshotOptions

Container for parameters required to configure the time of daily automated snapshots of the indices in an Amazon ES domain.

Field	Data Type	Description
AutomatedSnapshotStartHour	Integer	Hour during which the service takes an automated daily snapshot of the indices in the Amazon ES domain

SnapshotOptionsStatus

Status of an update to the configuration of the daily automated snapshot for an Amazon ES domain.

Field	Data Type	Description
Options	SnapshotOptions (p. 221)	Container for parameters required to configure the time of daily automated snapshots of indices in an Amazon ES domain
Status	OptionStatus (p. 220)	Status of an update to snapshot options for an Amazon ES domain

Tag

Field	Data Type	Description
Key	TagKey (p. 221)	Required name of the tag. Tag keys must be unique for the Amazon ES domain to which they are attached. For more information, see Tagging Amazon Elasticsearch Service Domains (p. 55).
Value	TagValue (p. 222)	Optional string value of the tag. Tag values can be null and do not have to be unique in a tag set. For example, you can have a key-value pair in a tag set of project/Trinity and cost-center/Trinity.

TagKey

Field	Data Type	Description
Key	String	Name of the tag. String can have up to 128 characters.

TagList

Field	Data Type	Description
Tag	Tag (p. 221)	Resource tag attached to an Amazon ES domain.

TagValue

Field	Data Type	Description
Value	String	Holds the value for a TagKey. String can have up to 256 characters.

VPCDerivedInfo

Field	Data Type	Description
VPCId	String	The ID for your VPC. Amazon VPC generates this value when you create a VPC.
SubnetIds	StringList	A list of subnet IDs associated with the VPC endpoints for the domain. To learn more, see VPCs and Subnets in the *Amazon VPC User Guide*.
AvailabilityZones	StringList	The list of Availability Zones associated with the VPC subnets. To learn more, see VPC and Subnet Basics in the *Amazon VPC User Guide*.
SecurityGroupIds	StringList	The list of security group IDs associated with the VPC endpoints for the domain. To learn more, see Security Groups for your VPC in the *Amazon VPC User Guide*.

VPCDerivedInfoStatus

Field	Data Type	Description
Options	VPCDerivedInfo (p. 222)	Information that Amazon ES derives based on VPCOptions (p. 223) for the domain.
Status	OptionStatus (p. 220)	Status of an update to VPC configuration options for an Amazon ES domain.

VPCOptions

Field	Data Type	Description
`SubnetIds`	StringList	A list of subnet IDs associated with the VPC endpoints for the domain. If your domain has zone awareness enabled, you need to provide two subnet IDs, one per zone. Otherwise, provide only one. To learn more, see VPCs and Subnets in the *Amazon VPC User Guide*.
`SecurityGroupIds`	StringList	The list of security group IDs associated with the VPC endpoints for the domain. If you do not provide a security group ID, Amazon ES uses the default security group for the VPC. To learn more, see Security Groups for your VPC in the *Amazon VPC User Guide*.

VPCOptionsStatus

Field	Data Type	Description
`Options`	`VPCOptions` (p. 223)	Container for the values required to configure Amazon ES to work with a VPC.
`Status`	`OptionStatus` (p. 220)	Status of an update to VPC configuration options for an Amazon ES domain.

Errors

Amazon ES throws the following errors:

Exception	Description
`BaseException`	Thrown for all service errors. Contains the HTTP status code of the error.
`ValidationException`	Thrown when the HTTP request contains invalid input or is missing required input. Returns HTTP status code 400.
`DisabledOperationException`	Thrown when the client attempts to perform an unsupported operation. Returns HTTP status code 409.
`InternalException`	Thrown when an error internal to the service occurs while processing a request. Returns HTTP status code 500.
`InvalidTypeException`	Thrown when trying to create or access an Amazon ES domain sub-resource that is either invalid or not supported. Returns HTTP status code 409.
`LimitExceededException`	Thrown when trying to create more than the allowed number and type of Amazon ES domain resources and sub-resources. Returns HTTP status code 409.

Exception	Description
`ResourceNotFoundException`	Thrown when accessing or deleting a resource that does not exist. Returns HTTP status code 400.
`ResourceAlreadyExistsException`	Thrown when a client attempts to create a resource that already exists in an Amazon ES domain. Returns HTTP status code 400.

Amazon Elasticsearch Service Limits

The following tables show limits for Amazon ES resources, including the number of instances per cluster, the minimum and maximum sizes for EBS volumes, and network limits.

Cluster and Instance Limits

The following table shows Amazon ES limits for clusters and instances.

Clusters and Instances	Limit
Maximum number of data instances (instance count) per cluster	20 (except for the T2 instance types, which have a maximum of 10) **Note** The default limit is 20 data instances per domain. To request an increase up to 100 per domain (for Elasticsearch 2.3 or later), create a case with the AWS Support Center. For more information about requesting an increase, see AWS Service Limits.
Maximum number of dedicated master nodes	5 **Note** You can use the T2 instance types as dedicated master nodes only if the instance count is 10 or fewer.
Smallest supported instance type	`t2.micro.elasticsearch` (versions 1.5 and 2.3) and `t2.small.elasticsearch` (version 5.*x* and 6.*x*).
Maximum number of domains per account (per region)	100

For a list of the instance types that Amazon ES supports, see Supported Instance Types (p. 150).

EBS Volume Size Limits

The following table shows the minimum and maximum sizes for EBS volumes for each instance type that Amazon ES supports. See Amazon Elasticsearch Service Pricing for information on which instance types offer instance storage.

> **Note**
> If you select magnetic storage under **EBS volume type** when creating your domain, maximum volume size is 100 GB for all instance types except `t2.micro`, `t2.small`, and `t2.medium`. For the maximum sizes listed in the following table, select one of the SSD options.

Instance Type	Minimum EBS Size	Maximum EBS Size
t2.micro.elasticsearch	10 GB	35 GB
t2.small.elasticsearch	10 GB	35 GB
t2.medium.elasticsearch	10 GB	35 GB
m3.medium.elasticsearch	10 GB	100 GB
m3.large.elasticsearch	10 GB	512 GB
m3.xlarge.elasticsearch	10 GB	512 GB
m3.2xlarge.elasticsearch	10 GB	512 GB
m4.large.elasticsearch	10 GB	512 GB
m4.xlarge.elasticsearch	10 GB	1 TB*
m4.2xlarge.elasticsearch	10 GB	1.5 TB*
m4.4xlarge.elasticsearch	10 GB	1.5 TB*
m4.10xlarge.elasticsearch	10 GB	1.5 TB*
c4.large.elasticsearch	10 GB	100 GB
c4.xlarge.elasticsearch	10 GB	512 GB
c4.2xlarge.elasticsearch	10 GB	1 TB*
c4.4xlarge.elasticsearch	10 GB	1.5 TB*
c4.8xlarge.elasticsearch	10 GB	1.5 TB*
r3.large.elasticsearch	10 GB	512 GB
r3.xlarge.elasticsearch	10 GB	512 GB
r3.2xlarge.elasticsearch	10 GB	512 GB
r3.4xlarge.elasticsearch	10 GB	512 GB
r3.8xlarge.elasticsearch	10 GB	512 GB
r4.large.elasticsearch	10 GB	1 TB*
r4.xlarge.elasticsearch	10 GB	1.5 TB*
r4.2xlarge.elasticsearch	10 GB	1.5 TB*
r4.4xlarge.elasticsearch	10 GB	1.5 TB*
r4.8xlarge.elasticsearch	10 GB	1.5 TB*
r4.16xlarge.elasticsearch	10 GB	1.5 TB*
i2.xlarge.elasticsearch	10 GB	512 GB
i2.2xlarge.elasticsearch	10 GB	512 GB
i3.large.elasticsearch	N/A	N/A

Instance Type	Minimum EBS Size	Maximum EBS Size
i3.xlarge.elasticsearch	N/A	N/A
i3.2xlarge.elasticsearch	N/A	N/A
i3.4xlarge.elasticsearch	N/A	N/A
i3.8xlarge.elasticsearch	N/A	N/A
i3.16xlarge.elasticsearch	N/A	N/A

* 512 GB is the maximum volume size that is supported with Elasticsearch version 1.5.

Network Limits

The following table shows the maximum size of HTTP request payloads.

Instance Type	Maximum Size of HTTP Request Payloads
t2.micro.elasticsearch	10 MB
t2.small.elasticsearch	10 MB
t2.medium.elasticsearch	10 MB
m3.medium.elasticsearch	10 MB
m3.large.elasticsearch	10 MB
m3.xlarge.elasticsearch	100 MB
m3.2xlarge.elasticsearch	100 MB
m4.large.elasticsearch	10 MB
m4.xlarge.elasticsearch	100 MB
m4.2xlarge.elasticsearch	100 MB
m4.4xlarge.elasticsearch	100 MB
m4.10xlarge.elasticsearch	100 MB
c4.large.elasticsearch	10 MB
c4.xlarge.elasticsearch	100 MB
c4.2xlarge.elasticsearch	100 MB
c4.4xlarge.elasticsearch	100 MB
c4.8xlarge.elasticsearch	100 MB
r3.large.elasticsearch	10 MB
r3.xlarge.elasticsearch	100 MB
r3.2xlarge.elasticsearch	100 MB

Instance Type	Maximum Size of HTTP Request Payloads
`r3.4xlarge.elasticsearch`	100 MB
`r3.8xlarge.elasticsearch`	100 MB
`r4.large.elasticsearch`	100 MB
`r4.xlarge.elasticsearch`	100 MB
`r4.2xlarge.elasticsearch`	100 MB
`r4.4xlarge.elasticsearch`	100 MB
`r4.8xlarge.elasticsearch`	100 MB
`r4.16xlarge.elasticsearch`	100 MB
`i2.xlarge.elasticsearch`	100 MB
`i2.2xlarge.elasticsearch`	100 MB
`i3.large.elasticsearch`	100 MB
`i3.xlarge.elasticsearch`	100 MB
`i3.2xlarge.elasticsearch`	100 MB
`i3.4xlarge.elasticsearch`	100 MB
`i3.8xlarge.elasticsearch`	100 MB
`i3.16xlarge.elasticsearch`	100 MB

Java Process Limit

Amazon ES limits Java processes to a heap size of 32 GB. Advanced users can specify the percentage of the heap used for field data. For more information, see the section called “Configuring Advanced Options” (p. 24) and the section called “JVM OutOfMemoryError” (p. 146).

Amazon Elasticsearch Service Reserved Instances

Amazon Elasticsearch Service Reserved Instances (RIs) offer significant discounts compared to standard On-Demand Instances. The instances themselves are identical; RIs are just a billing discount applied to On-Demand Instances in your account. For long-lived applications with predictable usage, RIs can provide considerable savings over time.

Amazon ES RIs require one- or three-year terms and have three payment options that affect the discount rate:

- **No Upfront** – You pay nothing upfront. You pay a discounted hourly rate for every hour within the term.
- **Partial Upfront** – You pay a portion of the cost upfront, and you pay a discounted hourly rate for every hour within the term.
- **All Upfront** – You pay the entirety of the cost upfront. You don't pay an hourly rate for the term.

Generally speaking, a larger upfront payment means a larger discount. You can't cancel Reserved Instances—when you reserve them, you commit to paying for the entire term—and upfront payments are nonrefundable. For full details, see Amazon Elasticsearch Service Pricing and FAQ.

Topics

Purchasing Reserved Instances (Console)

The console lets you view your existing Reserved Instances and purchase new ones.

To purchase a reservation

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. Choose **Reserved Instances**.

   On this page, you can view your existing reservations. If you have many reservations, you can filter them to more easily identify and view a particular reservation.

   > **Tip**
   > If you don't see the **Reserved Instances** link, create a domain (p. 10) in the region.
4. Choose **Purchase Reserved Instance**.
5. For **Reservation Name**, type a unique, descriptive name.
6. Choose an instance type, size, and number of instances. For guidance, see the section called “Sizing Amazon ES Domains” (p. 125).

7. Choose a term length and payment option.
8. Review the payment details carefully.
9. Choose **Submit**.
10. Review the purchase summary carefully. Purchases of Reserved Instances are non-refundable.
11. Choose **Purchase**.

Purchasing Reserved Instances (AWS CLI)

The AWS CLI has commands for viewing offerings, purchasing a reservation, and viewing your reservations. The following command and sample response show the offerings for a given AWS Region:

```
aws es describe-reserved-elasticsearch-instance-offerings --region us-east-1
{
  "ReservedElasticsearchInstanceOfferings": [
    {
      "FixedPrice": x,
      "ReservedElasticsearchInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "UsagePrice": 0.0,
      "PaymentOption": "PARTIAL_UPFRONT",
      "Duration": 31536000,
      "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",
      "CurrencyCode": "USD"
    }
  ]
}
```

For an explanation of each return value, see the following table.

Field	Description
`FixedPrice`	The upfront cost of the reservation.
`ReservedElasticsearchInstanceOfferingId`	The offering ID. Make note of this value if you want to reserve the offering.
`RecurringCharges`	The hourly rate for the reservation.
`UsagePrice`	A legacy field. For Amazon ES, this value is always 0.
`PaymentOption`	No Upfront, Partial Upfront, or All Upfront.
`Duration`	Length of the term in seconds: • 31536000 seconds is one year. • 94608000 seconds is three years.
`ElasticsearchInstanceType`	The instance type for the reservation. For information about the hardware resources that are allocated to each instance type, see Amazon Elasticsearch Service Pricing.

Field	Description
`CurrencyCode`	The currency for `FixedPrice` and `RecurringChargeAmount`.

This next example purchases a reservation:

```
aws es purchase-reserved-elasticsearch-instance-offering --reserved-elasticsearch-instance-
offering-id 1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a --reservation-name my-reservation --
instance-count 3 --region us-east-1
{
  "ReservationName": "my-reservation",
  "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a"
}
```

Finally, you can list your reservations for a given region using the following example:

```
aws es describe-reserved-elasticsearch-instances --region us-east-1
{
  "ReservedElasticsearchInstances": [
    {
      "FixedPrice": x,
      "ReservedElasticsearchInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "ReservationName": "my-reservation",
      "PaymentOption": "PARTIAL_UPFRONT",
      "UsagePrice": 0.0,
      "ReservedElasticsearchInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "State": "payment-pending",
      "StartTime": 1522872571.229,
      "ElasticsearchInstanceCount": 3,
      "Duration": 31536000,
      "ElasticsearchInstanceType": "m4.2xlarge.elasticsearch",
      "CurrencyCode": "USD"
    }
  ]
}
```

> **Note**
> `StartTime` is Unix epoch time, which is the number of seconds that have passed since midnight UTC of 1 January 1970. For example, 1522872571 epoch time is 20:09:31 UTC of 4 April 2018. You can use online converters.

To learn more about the commands used in the preceding examples, see the AWS CLI Command Reference.

Purchasing Reserved Instances (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the operations that are defined in the Amazon ES Configuration API Reference (p. 167), including the following:

- `DescribeReservedElasticsearchInstanceOfferings`

- `PurchaseReservedElasticsearchInstance`
- `DescribeReservedElasticsearchInstances`

For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Examining Costs

Cost Explorer is a free tool that you can use to view your spending data for the past 13 months. Analyzing this data helps you identify trends and understand if RIs fit your use case. If you already have RIs, you can group by **Purchase Option** and show amortized costs to compare that spending to your spending for On-Demand Instances. For more information, see Analyzing Your Costs with Cost Explorer in the *AWS Billing and Cost Management User Guide*.

Tutorial: Creating a Search Application with Amazon Elasticsearch Service

A common way to create a search application with Amazon ES is to use web forms to send user queries to a server. Then you can authorize the server to call the Elasticsearch APIs directly and have the server send requests to Amazon ES.

If you want to write client-side code that doesn't rely on a server, however, you should compensate for the security and performance risks. Allowing unsigned, public access to the Elasticsearch APIs is inadvisable. Users might access unsecured endpoints or impact cluster performance through overly broad queries (or too many queries).

This chapter presents a solution: use Amazon API Gateway to restrict users to a subset of the Elasticsearch APIs and AWS Lambda to sign requests from API Gateway to Amazon ES.

> **Note**
> Standard API Gateway and Lambda pricing applies, but within the limited usage of this tutorial, costs should be negligible.

Step 1: Index Sample Data

A prerequisite for these steps is an Amazon ES domain. Download sample-movies.zip, unzip it, and use the `_bulk` API to add the 5,000 documents to the `movies` index:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/_bulk
{ "index": { "_index": "movies", "_type": "movie", "_id": "tt1979320" } }
{"fields":{"directors":["Ron
 Howard"],"release_date":"2013-09-02T00:00:00Z","rating":8.3,"genres":
["Action","Biography","Drama","Sport"],"image_url":"http://ia.media-imdb.com/images/
M/MV5BMTQyMDE0MTY0OV5BMl5BanBnXkFtZTcwMjI2OTI0OQ@@._V1_SX400_.jpg","plot":"A re-
creation of the merciless 1970s rivalry between Formula One rivals James Hunt and Niki
 Lauda.","title":"Rush","rank":2,"running_time_secs":7380,"actors":["Daniel Brühl","Chris
 Hemsworth","Olivia Wilde"],"year":2013},"id":"tt1979320","type":"add"}
{ "index": { "_index": "movies", "_type": "movie", "_id": "tt1951264" } }
{"fields":{"directors":["Francis Lawrence"],"release_date":"2013-11-11T00:00:00Z","genres":
["Action","Adventure","Sci-Fi","Thriller"],"image_url":"http://ia.media-imdb.com/images/
M/MV5BMTAyMjQ3OTAxMzNeQTJeQWpwZ15BbWU4MDU0NzA1MzAx._V1_SX400_.jpg","plot":"Katniss
 Everdeen and Peeta Mellark become targets of the Capitol after their victory in the 74th
 Hunger Games sparks a rebellion in the Districts of Panem.","title":"The Hunger Games:
 Catching Fire","rank":4,"running_time_secs":8760,"actors":["Jennifer Lawrence","Josh
 Hutcherson","Liam Hemsworth"],"year":2013},"id":"tt1951264","type":"add"}
...
```

To learn more, see *Indexing Data* (p. 59).

Step 2: Create the API

Using API Gateway to create a more limited API simplifies the process of interacting with the Elasticsearch `_search` API. It also lets you enable security features like Amazon Cognito authentication and request throttling. Create and deploy an API according to the following table.

Setting	Values
API	Type: New API **Settings** API name: search-es-api Description: Public API for searching an Amazon Elasticsearch Service domain Endpoint type: Regional
Resource	/
HTTP Method	`GET`
Method Request	**Settings** Authorization: none Request validator: Validate query string parameters and headers API key required: false **URL Query String Parameters** Name: q Required: Yes
Integration Request	Integration type: Lambda function Use Lambda proxy integration: Yes Lambda Region: *`us-west-1`* Lambda function: search-es-lambda Invoke with caller credentials: No Credentials cache: Do not add caller credentials to cache key Use default timeout: Yes
Stage	Name: search-es-api-test **Default Method Throttling** Enable throttling: Yes Rate: 1000 Burst: 500

These settings configure an API that has only one method: a `GET` request to the endpoint root (`https://some-id.execute-api.us-west-1.amazonaws.com/search-es-api-test`). The request requires a single parameter (`q`), the query string to search for. When called, the method passes the request to Lambda, which executes the `search-es-lambda` function. For more information, see Creating an API in Amazon API Gateway and Deploying an API in Amazon API Gateway.

Step 3: Create the Lambda Function

In this solution, API Gateway passes requests to the following Python 2.7 Lambda function, which queries Amazon ES and returns results:

```
import boto3
import json
import requests
from requests_aws4auth import AWS4Auth

region = '' # For example, us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # For example, search-mydomain-id.us-west-1.es.amazonaws.com
index = 'movies'
url = 'https://' + host + '/' + index + '/_search'

# Lambda execution starts here
def handler(event, context):

    # Put the user query into the query DSL for more accurate search results.
    # Note that certain fields are boosted (^).
    query = {
        "size": 25,
        "query": {
            "multi_match": {
                "query": event['queryStringParameters']['q'],
                "fields": ["fields.title^4", "fields.plot^2", "fields.actors",
 "fields.directors"]
            }
        }
    }

    # ES 6.x requires an explicit Content-Type header
    headers = { "Content-Type": "application/json" }

    # Make the signed HTTP request
    r = requests.get(url, auth=awsauth, headers=headers, data=json.dumps(query))

    # Create the response and add some extra content to support CORS
    response = {
        "statusCode": 200,
        "headers": {
            "Access-Control-Allow-Origin": '*'
        },
        "isBase64Encoded": False
    }

    # Add the search results to the response
    response['body'] = r.text
    return response
```

The function must have the following trigger.

Trigger	API	Deployment Stage	Security
API Gateway	search-es-api	search-es-api-test	Open

For more information about creating Lambda functions and deployment packages, see Creating a Deployment Package (Python) in the *AWS Lambda Developer Guide* and the section called "Creating the Lambda Deployment Package" (p. 71) in this guide.

Step 4: Modify the Domain Access Policy

Your Amazon ES domain must allow the Lambda function to make `GET` requests to the `movies` index. The following policy provides `search-es-role` (created through Lambda) access to the `movies` index:

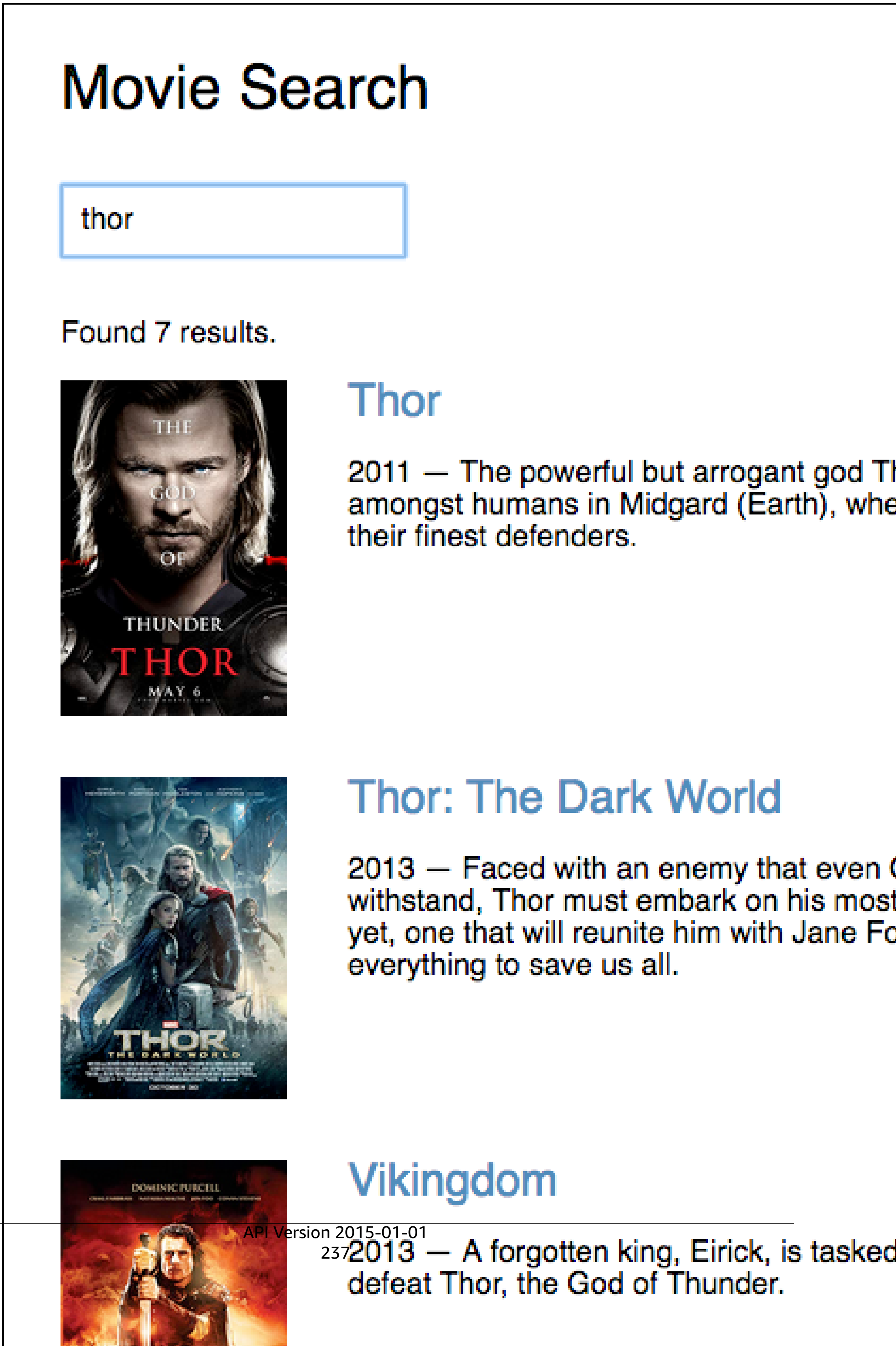
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/service-role/search-es-role"
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/web/movies/_search"
    }
  ]
}
```

For more information, see the section called "Configuring Access Policies" (p. 21).

Step 5: Test the Web Application

To test the web application

1. Download sample-site.zip, unzip it, and open `scripts/search.js` in your favorite text editor.
2. Update the `apigatewayendpoint` variable to point to your API Gateway endpoint. The endpoint takes the form of `https://`*`some-id`*`.execute-api.`*`us-west-1`*`.amazonaws.com/search-es-api-test`.
3. Open `index.html` and try running searches for *thor*, *house*, and a few other terms.



Movie Search

thor

Found 7 results.

Thor

2011 — The powerful but arrogant god Th
amongst humans in Midgard (Earth), whe
their finest defenders.

Thor: The Dark World

2013 — Faced with an enemy that even (
withstand, Thor must embark on his most
yet, one that will reunite him with Jane Fo
everything to save us all.

Vikingdom

2013 — A forgotten king, Eirick, is tasked
defeat Thor, the God of Thunder.

Next Steps

This chapter is just a starting point to demonstrate a concept. You might consider the following modifications:

- Add your own data to the Amazon ES domain.
- Add methods to your API.
- In the Lambda function, modify the search query or boost different fields.
- Style the results differently or modify `search.js` to display different fields to the user.

Tutorial: Visualizing Customer Support Calls with Amazon Elasticsearch Service and Kibana

This chapter is a full walkthrough of the following situation: a business receives some number of customer support calls and wants to analyze them. What is the subject of each call? How many were positive? How many were negative? How can managers search or review the the transcripts of these calls?

A manual workflow might involve employees listening to recordings, noting the subject of each call, and deciding whether or not the customer interaction was positive.

Such a process would be extremely labor-intensive. Assuming an average time of 10 minutes per call, each employee could listen to only 48 calls per day. Barring human bias, the data they generate would be highly accurate, but the *amount* of data would be minimal: just the subject of the call and a Boolean for whether or not the customer was satisfied. Anything more involved, such as a full transcript, would take a huge amount of time.

Using Amazon S3, Amazon Transcribe, Amazon Comprehend, and Amazon Elasticsearch Service (Amazon ES), you can automate a similar process with very little code and end up with much more data. For example, you can get a full transcript of the call, keywords from the transcript, and an overall "sentiment" of the call (positive, negative, neutral, or mixed). Then you can use Elasticsearch and Kibana to search and visualize the data.

While you can use this walkthrough as-is, the intent is to spark ideas about how to enrich your JSON documents before you index them in Amazon ES.

Estimated Costs

In general, performing the steps in this walkthrough should cost less than $2. The walkthrough uses the following resources:

- S3 bucket with less than 100 MB transferred and stored

  To learn more, see Amazon S3 Pricing.
- Amazon ES domain with one `t2.medium` instance and 10 GB of EBS storage for several hours

  To learn more, see Amazon Elasticsearch Service Pricing.
- Several calls to Amazon Transcribe

  To learn more, see Amazon Transcribe Pricing.
- Several natural language processing calls to Amazon Comprehend

  To learn more, see Amazon Comprehend Pricing.

Topics

- Step 2: Copy Sample Code (p. 240)
- (Optional) Step 3: Add Sample Data (p. 243)
- Step 4: Analyze and Visualize Your Data (p. 244)
- Step 5: Clean Up Resources and Next Steps (p. 252)

Step 1: Configure Prerequisites

Before proceeding, you must have the following resources.

Prerequisite	Description
Amazon S3 Bucket	For more information, see Creating a Bucket in the *Amazon Simple Storage Service Getting Started Guide*.
Amazon ES Domain	The destination for data. For more information, see Creating Amazon ES Domains (p. 10).

If you don't already have these resources, you can create them using the following AWS CLI commands:

```
aws s3 mb s3://my-transcribe-test --region us-west-2
```

```
aws es create-elasticsearch-domain --domain-name my-transcribe-test --elasticsearch-version
 6.2 --elasticsearch-cluster-config  InstanceType=t2.medium.elasticsearch,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 --access-
policies '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
{"AWS":"arn:aws:iam::123456789012:root"},"Action":"es:*","Resource":"arn:aws:es:us-
west-2:123456789012:domain/my-transcribe-test/*"}]}' --region us-west-2
```

Note
These commands use the `us-west-2` region, but you can use any region that Amazon Comprehend supports. To learn more, see the AWS General Reference.

Step 2: Copy Sample Code

1. Copy and paste the following sample code into a new file named `call-center.py`:

```
import boto3
import datetime
import json
import requests
from requests_aws4auth import AWS4Auth
import time
import urllib2

# Variables to update
audio_file_name = '' # For example, 000001.mp3
bucket_name = '' # For example, my-transcribe-test
domain = '' # For example, https://search-my-transcribe-test-12345.us-
west-2.es.amazonaws.com
index = 'support-calls'
type = 'call'
```

```
es_region = 'us-west-2'

# Upload audio file to S3.
s3_client = boto3.client('s3')

audio_file = open(audio_file_name, 'r')

print('Uploading ' + audio_file_name + '...')
response = s3_client.put_object(
    Body=audio_file,
    Bucket=bucket_name,
    Key=audio_file_name,
)

response = s3_client.get_bucket_location(
    Bucket=bucket_name
)

bucket_region = response['LocationConstraint']

# Build the URL to the audio file on S3.
mp3_uri = 'https://s3-' + bucket_region + '.amazonaws.com/' + bucket_name + '/' +
 audio_file_name

# Start transcription job.
transcribe_client = boto3.client('transcribe')

print('Starting transcription job...')
response = transcribe_client.start_transcription_job(
    TranscriptionJobName=audio_file_name,
    LanguageCode='en-US',
    MediaFormat='mp3',
    Media={
        'MediaFileUri': mp3_uri
    },
    Settings={
        'ShowSpeakerLabels': True,
        'MaxSpeakerLabels': 2 # assumes two people on a phone call
    }
)

# Wait for the transcription job to finish.
print('Waiting for job to complete...')
while True:
    response =
 transcribe_client.get_transcription_job(TranscriptionJobName=audio_file_name)
    if response['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    else:
        print('Still waiting...')
    time.sleep(10)

transcript_uri = response['TranscriptionJob']['Transcript']['TranscriptFileUri']

# Open the JSON file, read it, and get the transcript.
response = urllib2.urlopen(transcript_uri)
raw_json = response.read()
loaded_json = json.loads(raw_json)
transcript = loaded_json['results']['transcripts'][0]['transcript']

# Send transcript to Comprehend for key phrases and sentiment.
comprehend_client = boto3.client('comprehend')

# If necessary, trim the transcript.
# If the transcript is more than 5 KB, the Comprehend calls fail.
```

```
if len(transcript) > 5000:
    trimmed_transcript = transcript[:5000]
else:
    trimmed_transcript = transcript

print('Detecting key phrases...')
response = comprehend_client.detect_key_phrases(
    Text=trimmed_transcript,
    LanguageCode='en'
)

keywords = []
for keyword in response['KeyPhrases']:
    keywords.append(keyword['Text'])

print('Detecting sentiment...')
response = comprehend_client.detect_sentiment(
    Text=trimmed_transcript,
    LanguageCode='en'
)

sentiment = response['Sentiment']

# Build the Amazon Elasticsearch Service URL.
id = audio_file_name.strip('.mp3')
url = domain + '/' + index + '/' + type + '/' + id

# Create the JSON document.
json_document = {'transcript': transcript, 'keywords': keywords, 'sentiment':
 sentiment, 'timestamp': datetime.datetime.now().isoformat()}

# Provide all details necessary to sign the indexing request.
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, es_region, 'es',
 session_token=credentials.token)

# Add explicit header for Elasticsearch 6.x.
headers = {'Content-Type': 'application/json'}

# Index the document.
print('Indexing document...')
response = requests.put(url, auth=awsauth, json=json_document, headers=headers)

print(response)
print(response.json())
```

2. Update the initial six variables.
3. Install the required packages using the following commands:

```
pip install boto3
pip install requests
pip install requests_aws4auth
```

4. Place your MP3 in the same directory as `call-center.py` and run the script. A sample output follows:

```
$ python call-center.py
Uploading 000001.mp3...
Starting transcription job...
Waiting for job to complete...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Detecting key phrases...
Detecting sentiment...
Indexing document...
<Response [201]>
{u'_type': u'call', u'_seq_no': 0, u'_shards': {u'successful': 1, u'failed': 0,
 u'total': 2}, u'_index': u'support-calls4', u'_version': 1, u'_primary_term': 1,
 u'result': u'created', u'_id': u'000001'}
```

`call-center.py` performs a number of operations:

1. The script uploads an audio file (in this case, an MP3, but Amazon Transcribe supports several formats) to your S3 bucket.
2. It sends the audio file's URL to Amazon Transcribe and waits for the transcription job to finish.

   The time to finish the transcription job depends on the length of the audio file. Assume minutes, not seconds.

   Tip
   To improve the quality of the transcription, you can configure a custom vocabulary for Amazon Transcribe.
3. After the transcription job finishes, the script extracts the transcript, trims it to 5,000 characters, and sends it to Amazon Comprehend for keyword and sentiment analysis.
4. Finally, the script adds the full transcript, keywords, sentiment, and current time stamp to a JSON document and indexes it in Amazon ES.

Tip
LibriVox has public domain audiobooks that you can use for testing.

(Optional) Step 3: Add Sample Data

If you don't have a bunch of call recordings handy—and who does?—you can index (p. 59) the sample documents in sample-calls.zip, which are comparable to what `call-center.py` produces.

1. Create a file named `bulk-helper.py`:

```
import boto3
from elasticsearch import Elasticsearch, RequestsHttpConnection
import json
from requests_aws4auth import AWS4Auth

host = '' # For example, my-test-domain.us-west-2.es.amazonaws.com
region = '' # For example, us-west-2
service = 'es'

bulk_file = open('sample-calls.bulk', 'r').read()

credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

es = Elasticsearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    connection_class = RequestsHttpConnection
)

response = es.bulk(bulk_file)
print(json.dumps(response, indent=2, sort_keys=True))
```

2. Update the initial two variables for `host` and `region`.
3. Install the required package using the following command:

```
pip install elasticsearch
```

4. Download and unzip sample-calls.zip.
5. Place `sample-calls.bulk` in the same directory as `bulk-helper.py` and run the helper. A sample output follows:

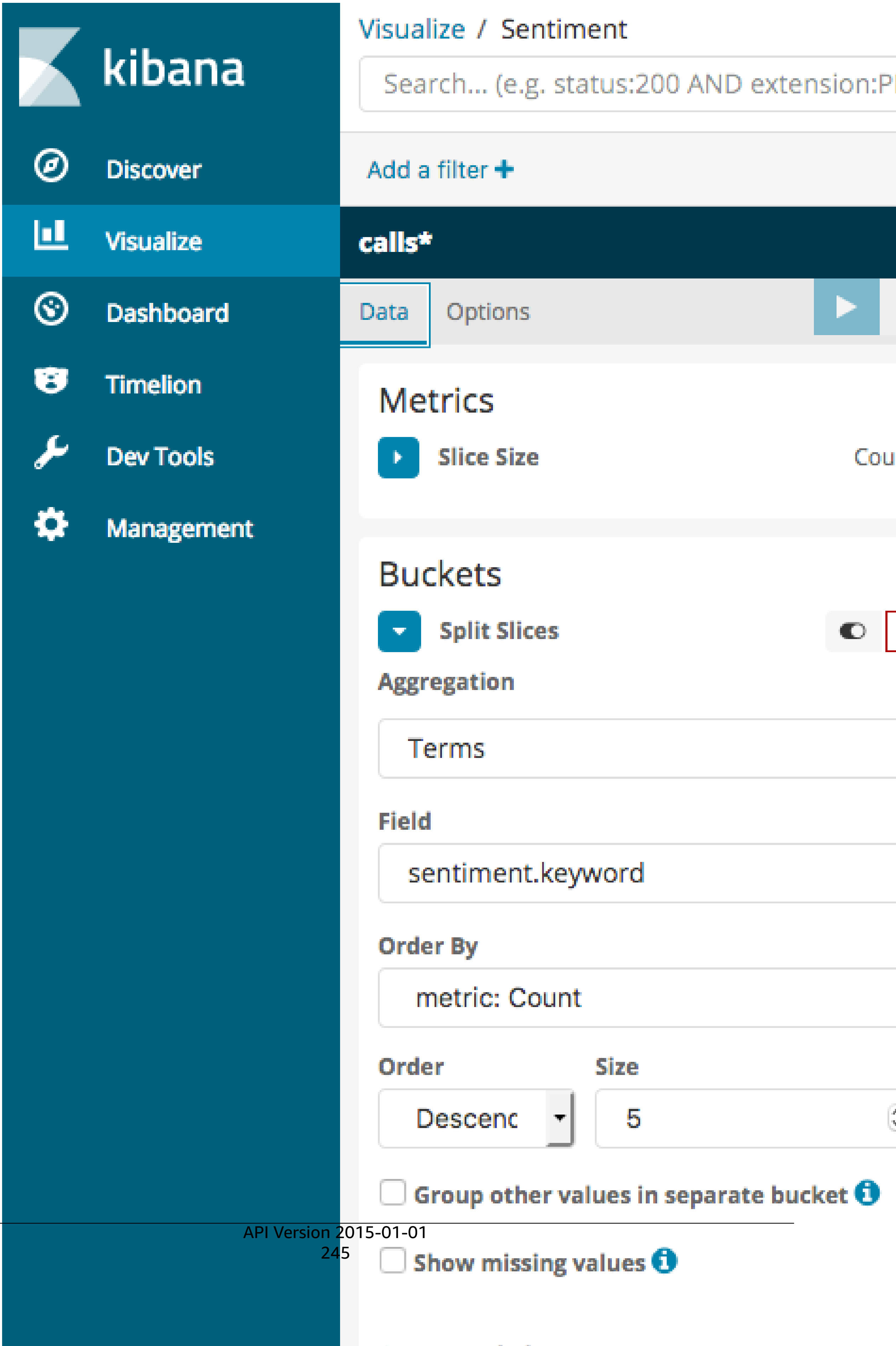
```
$ python bulk-helper.py
{
  "errors": false,
  "items": [
    {
      "index": {
        "_id": "1",
        "_index": "test-data",
        "_primary_term": 1,
        "_seq_no": 42,
        "_shards": {
          "failed": 0,
          "successful": 1,
          "total": 2
        },
        "_type": "call",
        "_version": 9,
        "result": "updated",
        "status": 200
      }
    },
    ...
  ],
  "took": 27
}
```

Step 4: Analyze and Visualize Your Data

Now that you have some data in Amazon ES, you can visualize it using Kibana.

1. Navigate to `https://search-`*`domain`*`.`*`region`*`.es.amazonaws.com/_plugin/kibana`.
2. Before you can use Kibana, you need an index pattern. Kibana uses index patterns to narrow your analysis to one or more indices. To match the `support-calls` index that `call-center.py` created, define an index pattern of `support*`, and then choose **Next step**.
3. For **Time Filter field name**, choose **timestamp**.
4. Now you can start creating visualizations. Choose **Visualize**, and then add a new visualization.
5. Choose the pie chart and the `support*` index pattern.
6. The default visualization is basic, so choose **Split Slices** to create a more interesting visualization.

   For **Aggregation**, choose **Terms**. For **Field**, choose **sentiment.keyword**. Then choose **Apply changes** and **Save**.



Metrics

Slice Size Cou

Buckets

Split Slices

Aggregation

Terms

Field

sentiment.keyword

Order By

metric: Count

Order Descenc

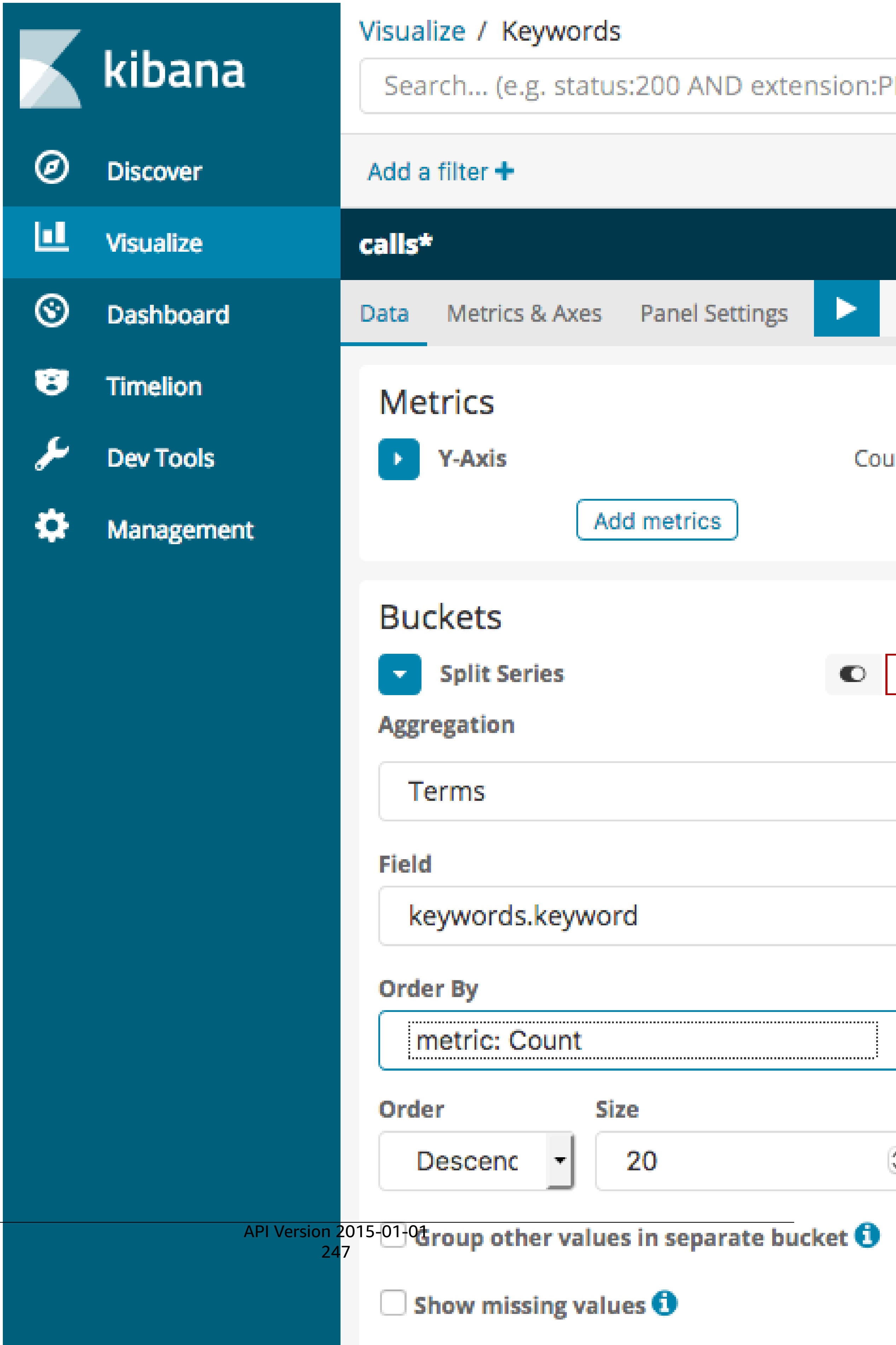
Size 5

□ Group other values in separate bucket

□ Show missing values

7. Return to the **Visualize** page, and add another visualization. This time, choose the horizontal bar chart.
8. Choose **Split Series**.

   For **Aggregation**, choose **Terms**. For **Field**, choose **keywords.keyword** and change **Size** to 20. Then choose **Apply Changes** and **Save**.



Aggregation

Terms

Field

keywords.keyword

Order By

metric: Count

Order

Descenc

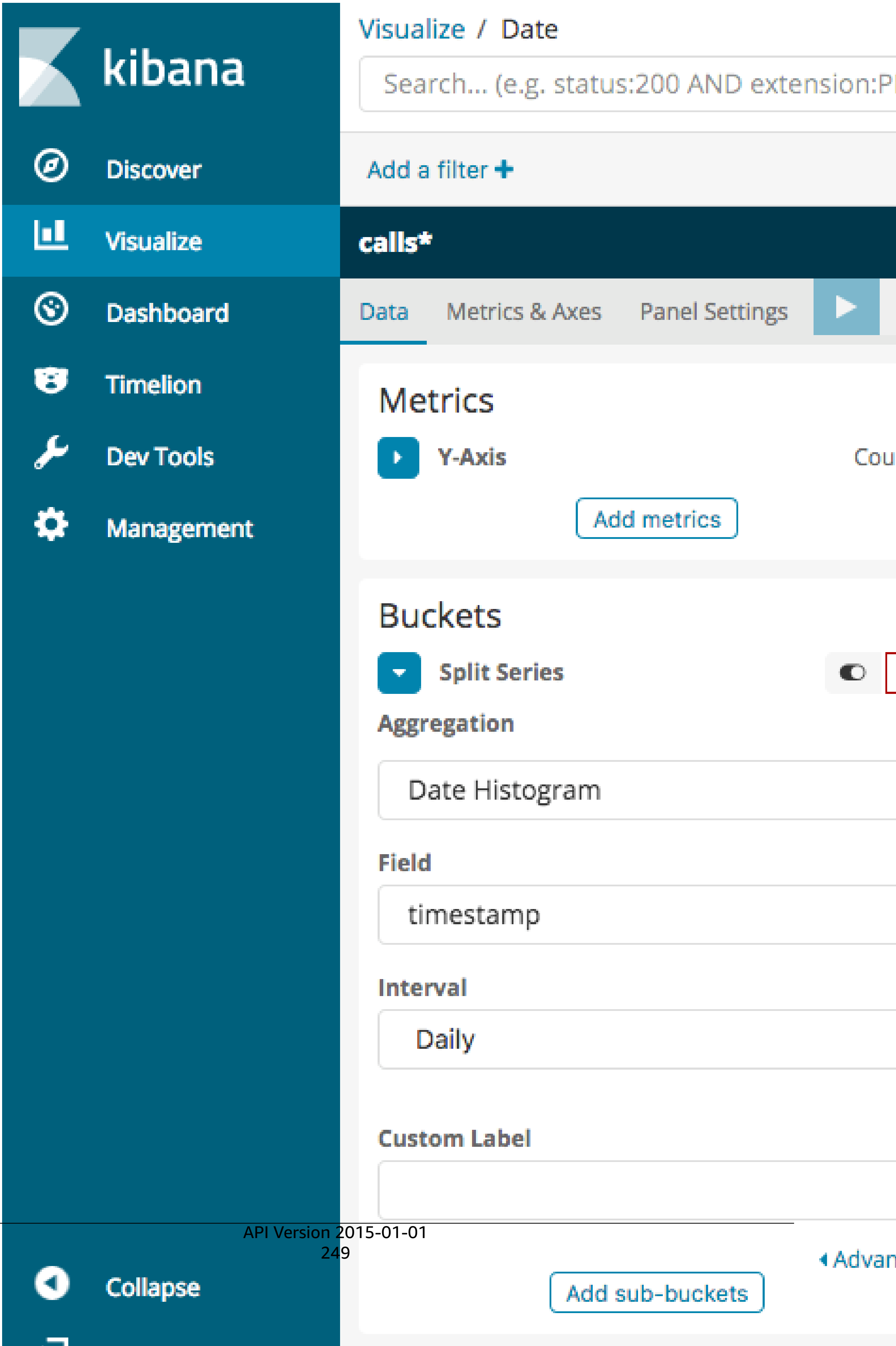
Size

20

□ Group other values in separate bucket

□ Show missing values

9. Return to the **Visualize** page and add one final visualization, a vertical bar chart.
10. Choose **Split Series**. For **Aggregation**, choose **Date Histogram**. For **Field**, choose **timestamp** and change **Interval** to **Daily**.
11. Choose **Metrics & Axes** and change **Mode** to **normal**.
12. Choose **Apply Changes** and **Save**.



Aggregation

Date Histogram

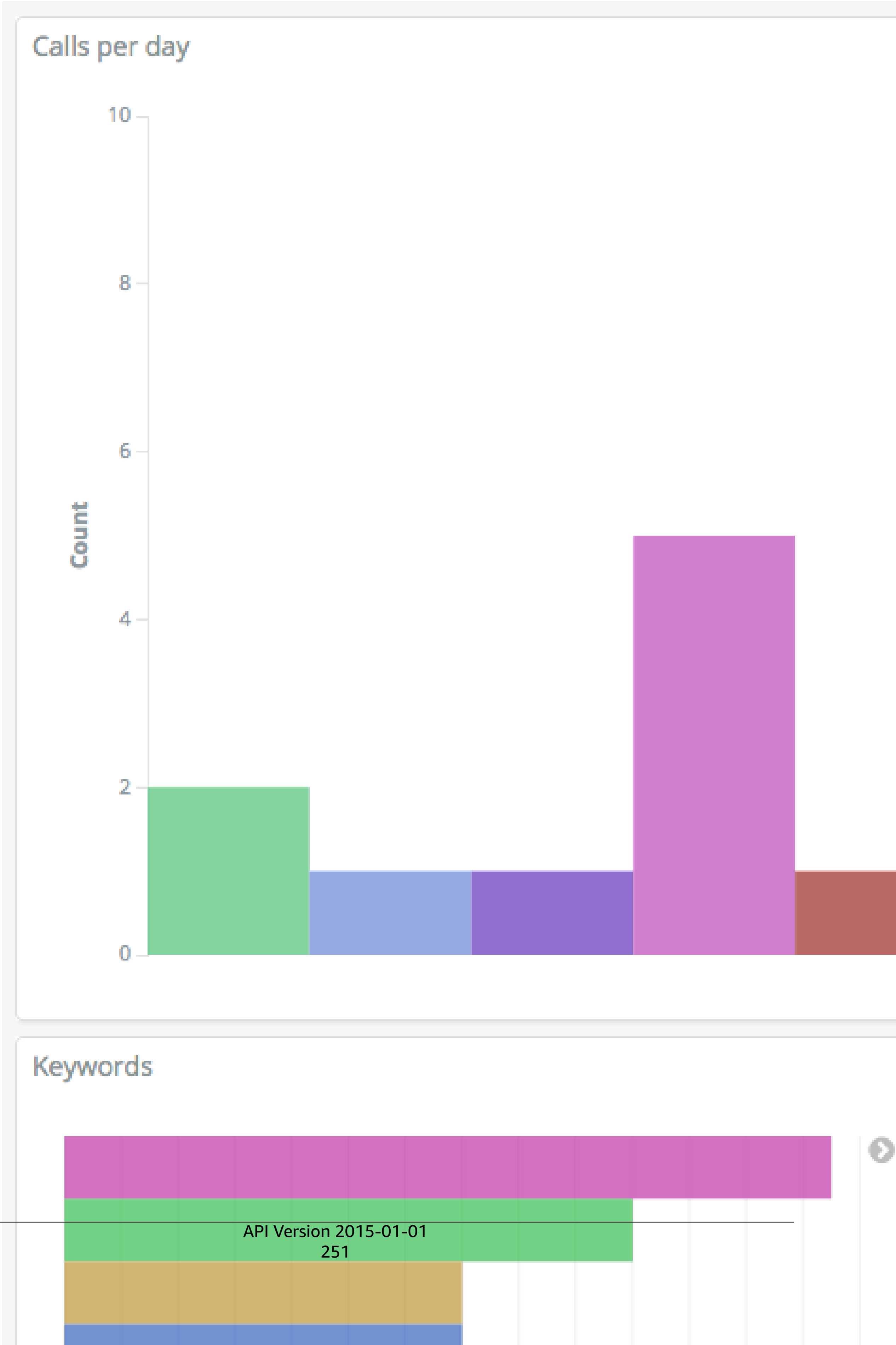
Field

timestamp

Interval

Daily

13. Now that you have three visualizations, you can add them to a Kibana dashboard. Choose **Dashboard**, create a dashboard, and add your visualizations.


Calls per day
10
8
6
4
2
0
Count
Keywords

Step 5: Clean Up Resources and Next Steps

To avoid unnecessary charges, delete the S3 bucket and Amazon ES domain. To learn more, see Delete a Bucket in the *Amazon Simple Storage Service Developer Guide* and Delete an Amazon ES Domain (p. 9) in this guide.

Transcripts require much less disk space than MP3 files. You might be able to shorten your MP3 retention window—for example, from three months of call recordings to one month—retain years of transcripts, and still save on storage costs.

You could also automate the transcription process using AWS Step Functions and Lambda, add additional metadata before indexing, or craft more complex visualizations to fit your exact use case.

Document History for Amazon Elasticsearch Service

This topic describes important changes to Amazon Elasticsearch Service (Amazon ES).

Relevant Dates to this History:

- **Current product version—**2015-01-01
- **Latest product release—**17 October 2018
- **Latest documentation update—**17 October 2018

Change	Description	Release Date
China (Beijing) Support	Amazon Elasticsearch Service is now available in the cn-north-1 region and supports the M4, C4, and R4 instance types.	17 October 2018
Node-to-node Encryption	Amazon Elasticsearch Service now supports node-to-node encryption, which keeps your data encrypted as Elasticsearch distributes it throughout your cluster. To learn more, see the section called "Node-to-node Encryption" (p. 136).	18 September 2018
Elasticsearch 6.3 and 5.6 Support	Amazon Elasticsearch Service now supports Elasticsearch version 6.3 and 5.6. To learn more, see the section called "Supported Elasticsearch Versions" (p. 2).	14 August 2018
In-place Version Upgrades	Amazon Elasticsearch Service now supports in-place version upgrades for Elasticsearch. To learn more, see *Upgrading Elasticsearch* (p. 94).	14 August 2018
Error Logs	Amazon ES now supports the publishing of Elasticsearch error logs to CloudWatch Logs. To learn more, see the section called "Configuring Logs" (p. 25).	31 July 2018
China (Ningxia) Reserved Instances	Amazon ES now offers Reserved Instances in the China (Ningxia) region.	29 May 2018
Reserved Instances	Amazon ES now offers Reserved Instances. To learn more, see *Reserved Instances* (p. 229).	7 May 2018
Amazon Cognito Authentication for Kibana	Amazon ES now offers login page protection for Kibana. To learn more, see *Authentication for Kibana* (p. 104).	2 April 2018
Elasticsearch 6.2 Support	Amazon Elasticsearch Service now supports Elasticsearch version 6.2.	14 March 2018
Korean Analysis Plugin	Amazon ES now supports a memory-optimized version of the Seunjeon Korean analysis plugin.	13 March 2018
Instant Access Control Updates	Changes to the access control policies on Amazon ES domains now take effect instantly.	7 March 2018

Change	Description	Release Date
Petabyte Scale	Amazon ES now supports I3 instance types and total domain storage of up to 1.5 PB. To learn more, see *Petabyte Scale* (p. 132).	19 December 2017
Encryption of Data at Rest	Amazon ES now supports encryption of data at rest. To learn more, see the section called "Encryption at Rest" (p. 134).	7 December 2017
Elasticsearch 6.0 Support	Amazon ES now supports Elasticsearch version 6.0. For migration considerations and instructions, see *Upgrading Elasticsearch* (p. 94).	6 December 2017
VPC Support	Amazon ES now lets you launch domains within an Amazon Virtual Private Cloud. VPC support provides an additional layer of security and simplifies communications between Amazon ES and other services within a VPC. To learn more, see *VPC Support* (p. 117).	17 October 2017
Slow Logs Publishing	Amazon ES now supports the publishing of slow logs to CloudWatch Logs. To learn more, see the section called "Configuring Logs" (p. 25).	16 October 2017
Elasticsearch 5.5 Support	Amazon ES now supports Elasticsearch version 5.5. For new feature summaries, see the Amazon announcement of availability. You can now restore automated snapshots without contacting AWS Support and store scripts using the Elasticsearch `_scripts` API.	7 September 2017
Elasticsearch 5.3 Support	Amazon ES added support for Elasticsearch version 5.3.	1 June 2017
More Instances and EBS Capacity per Cluster	Amazon ES now supports up to 100 nodes and 150 TB EBS capacity per cluster.	5 April 2017
Canada (Central) and EU (London) Support	Amazon ES added support for the following regions: Canada (Central), ca-central-1, and EU (London), eu-west-2.	20 March 2017
More Instances and Larger EBS Volumes	Amazon ES added support for more instances and larger EBS volumes.	21 February 2017
Elasticsearch 5.1 Support	Amazon ES added support for Elasticsearch version 5.1.	30 January 2017
Support for the Phonetic Analysis Plugin	Amazon ES now provides built-in integration with the Phonetic Analysis plugin, which allows you to run "sounds-like" queries on your data.	22 December 2016
US East (Ohio) Support	Amazon ES added support for the following region: US East (Ohio), `us-east-2`.	17 October 2016
New Performance Metric	Amazon ES added a performance metric, `ClusterUsedSpace`.	29 July 2016
Elasticsearch 2.3 Support	Amazon ES added support for Elasticsearch version 2.3.	27 July 2016

Change	Description	Release Date
Asia Pacific (Mumbai) Support	Amazon ES added support for the following region: Asia Pacific (Mumbai), ap-south-1.	27 June 2016
More Instances per Cluster	Amazon ES increased the maximum number of instances (instance count) per cluster from 10 to 20.	18 May 2016
Asia Pacific (Seoul) Support	Amazon ES added support for the following region: Asia Pacific (Seoul), ap-northeast-2.	28 January 2016
Amazon ES	Initial release.	1 October 2015

Using Service-Linked Roles for Amazon ES

Amazon Elasticsearch Service uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Amazon ES. Service-linked roles are predefined by Amazon ES and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up Amazon ES easier because you don’t have to manually add the necessary permissions. Amazon ES defines the permissions of its service-linked roles, and unless defined otherwise, only Amazon ES can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting its related resources. This protects your Amazon ES resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-Linked Role Permissions for Amazon ES

Amazon ES uses the service-linked role named **AWSServiceRoleForAmazonElasticsearchService**.

The AWSServiceRoleForAmazonElasticsearchService service-linked role trusts the following services to assume the role:

- `es.amazonaws.com`

The role permissions policy allows Amazon ES to complete the following actions on the specified resources:

- Action: `ec2:CreateNetworkInterface` on `*`
- Action: `ec2:DeleteNetworkInterface` on `*`
- Action: `ec2:DescribeNetworkInterfaces` on `*`
- Action: `ec2:ModifyNetworkInterfaceAttribute` on `*`
- Action: `ec2:DescribeSecurityGroups` on `*`
- Action: `ec2:DescribeSubnets` on `*`

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a Service-Linked Role for Amazon ES

You don't need to manually create a service-linked role. When you create a VPC access domain using the AWS Management Console, Amazon ES creates the service-linked role for you. In order for this automatic creation to succeed, you must have permissions for the `iam:CreateServiceLinkedRole` action.

If you delete this service-linked role and then need to create it again, you can use the same process to recreate the role in your account.

You can also use the IAM console, the IAM CLI, or the IAM API to create a service-linked role manually. For more information, see Creating a Service-Linked Role in the *IAM User Guide*.

Editing a Service-Linked Role for Amazon ES

Amazon ES does not allow you to edit the AWSServiceRoleForAmazonElasticsearchService service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a Service-Linked Role for Amazon ES

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way you don’t have an unused entity that is not actively monitored or maintained. However, you must clean up your service-linked role before you can manually delete it.

Cleaning Up a Service-Linked Role

Before you can use IAM to delete a service-linked role, you must first confirm that the role has no active sessions and remove any resources used by the role.

To check whether the service-linked role has an active session in the IAM console

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane of the IAM console, choose **Roles**. Then choose the name (not the check box) of the AWSServiceRoleForAmazonElasticsearchService role.
3. On the **Summary** page for the selected role, choose the **Access Advisor** tab.
4. On the **Access Advisor** tab, review recent activity for the service-linked role.

   Note
   If you are unsure whether Amazon ES is using the AWSServiceRoleForAmazonElasticsearchService role, you can try to delete the role. If the service is using the role, then the deletion fails and you can view the regions where the role is being used. If the role is being used, then you must wait for the session to end before you can delete the role. You cannot revoke the session for a service-linked role.

To remove Amazon ES resources used by the AWSServiceRoleForAmazonElasticsearchService

1. Sign in to the AWS Management Console and open the Amazon ES console.
2. Delete any domains that list **VPC** under the **Endpoint** column.

Manually Delete a Service-Linked Role

Use the Amazon ES configuration API to delete the AWSServiceRoleForAmazonElasticsearchService service-linked role. For more information, see the section called “DeleteElasticsearchServiceRole” (p. 176).

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.