Amazon Elasticsearch Service

Developer Guide

API Version 2015-01-01

Amazon Elasticsearch Service: Developer Guide

Copyright © 2016 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is Amazon Elasticsearch Service? 1
- Features of Amazon Elasticsearch Service 2
- How to Get Started with Amazon Elasticsearch Service 2
  - Signing Up for AWS 3
  - Accessing Amazon Elasticsearch Service 3
  - Regions and Endpoints 3
  - Scaling 4
  - Signing Requests 5
  - Choosing an Instance Type 5
  - Using Amazon EBS Volumes for Storage 6
  - Choosing an Elasticsearch Version 6
  - Related Services 6
  - Pricing for Amazon Elasticsearch Service 7

Getting Started with Amazon ES Domains 8
- Step 1: Signing Up for AWS 9
- Step 2: Creating an Amazon ES Domain 9
- Step 3: Configuring Access Policies for a Domain 11
- Step 4: Uploading Data for Indexing 12
- Step 5: Searching Documents from an Amazon ES Domain 13
- Step 6: Deleting an Amazon ES Domain 13

Creating and Configuring Amazon ES Domains 15
- Creating Amazon ES Domains 15
  - Creating Amazon ES Domains (Console) 15
  - Creating Amazon ES Domains (AWS CLI) 18
  - Creating Amazon ES Domains (AWS SDKs) 20
- Configuring Amazon ES Domains 20
  - Configuring Amazon ES Domains (Console) 21
  - Configuring Amazon ES Domains (AWS CLI) 22
  - Configuring Amazon ES Domains (AWS SDKs) 23
- Configuring EBS-based Storage 23
  - Configuring EBS-based Storage (Console) 24
  - Configuring EBS-based Storage (AWS CLI) 24
  - Configuring EBS-based Storage (AWS SDKs) 26
- Configuring Access Policies 26
  - Configuring Access Policies (Console) 27
  - Configuring Access Policies (AWS CLI) 28
  - Configuring Access Policies (AWS SDKs) 31
- Configuring Snapshots 31
  - Configuring Snapshots (Console) 31
  - Configuring Snapshots (AWS CLI) 31
  - Configuring Snapshots (AWS SDKs) 32
- Configuring Advanced Options 32
  - Configuring Advanced Options (Console) 32
  - Configuring Advanced Options (AWS CLI) 33
  - Configuring Advanced Options (AWS SDKs) 33

Managing Amazon ES Domains 34
- About Dedicated Master Nodes 34
- Enabling Zone Awareness (Console) 36
- Working with Manual Index Snapshots (AWS CLI) 37
  - Snapshot Prerequisites 38
  - Registering a Snapshot Directory 40
  - Taking Manual Snapshots (AWS CLI) 41
  - Restoring Manual Snapshots (AWS CLI) 41
- Monitoring Cluster Metrics and Statistics with Amazon CloudWatch (Console) 42
  - Cluster Metrics 43

Dedicated Master Node Metrics 46
EBS Volume Metrics 47
Logging Configuration Service Calls Using AWS CloudTrail 49
Amazon ES Information in CloudTrail 50
Understanding Amazon ES Log File Entries 50
Visualizing Data with Kibana 51
Loading Bulk Data with the Logstash Plugin 51
Signing Amazon Elasticsearch Service Requests 53
Tagging Amazon Elasticsearch Service Domains 54
Working with Tags (Console) 54
Working with Tags (AWS CLI) 55
Working with Tags (AWS SDKs) 56
Loading Streaming Data into Amazon ES 57
Loading Streaming Data into Amazon ES from Amazon S3 58
Setting Up to Load Streaming Data into Amazon ES from Amazon S3 58
Loading Streaming Data into Amazon Elasticsearch Service from Amazon Kinesis 61
Setting Up to Load Streaming Data into Amazon ES from Amazon Kinesis 62
Loading Streaming Data into Amazon ES from Amazon DynamoDB 64
Loading Streaming Data into Amazon ES from Amazon CloudWatch 64
Migrating from Version 1.5 to Version 2.3 65
Supported Elasticsearch Operations 67
Handling AWS Service Errors 69
Failed Cluster Nodes 69
Red Cluster Status 69
Recovering from a Lack of Free Storage Space 70
Recovering from a Continuous Heavy Processing Load 70
Yellow Cluster Status 71
ClusterBlockException 71
Logstash with Zero FreeStorageSpace 71
Block Disks due to Low Memory 71
JVM OutOfMemoryError 71
Troubleshooting 73
Kibana: I Cannot Sign AWS Service Requests to the Kibana Service Endpoint 73
Kibana: I Don't See the Indices for My Elasticsearch Domain in Kibana 4 74
Kibana: I Get a Browser Error When I Use Kibana to View My Data 75
Integrations: I Don't See a Service Role for Amazon ES in the IAM Console 76
Amazon ES Configuration API Reference 77
Actions 77
AddTags 78
CreateElasticsearchDomain 79
DeleteElasticsearchDomain 82
DescribeElasticsearchDomain 84
DescribeElasticsearchDomains 85
ListDomainNames 88
DescribeElasticsearchDomainConfig 89
ListTags 91
RemoveTags 92
UpdateElasticsearchDomainConfig 93
Data Types 97
AdvancedOptions 97
AdvancedOptionsStatus 97
ARN 98
CreateElasticsearchDomainRequest 98
DomainID 99
DomainName 99
DomainNameList 99
EBSOptions 99
ElasticsearchClusterConfig 100

ElasticsearchDomainConfig 100
ElasticsearchDomainStatus 101
ElasticsearchDomainStatusList 103
OptionState 103
OptionStatus 103
ServiceURL 104
SnapshotOptions 104
SnapshotOptionsStatus 104
Tag 105
TagKey 105
TagList 105
TagValue 105
Errors 105
Limits 107
Document History for Amazon ES 110
AWS Glossary 112

What Is Amazon Elasticsearch Service?

Amazon Elasticsearch Service (Amazon ES) is a managed service that makes it easy to create a domain and 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 analytics. With Amazon ES, you get direct access to Elasticsearch open-source APIs so that existing code and applications work seamlessly together. Currently, Amazon ES supports Elasticsearch versions 1.5 and 2.3. To learn more about Elasticsearch and its uses, see Getting Started in the *Elasticsearch Reference*.

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

To get started using the service, you create an Amazon ES domain. An Amazon ES domain is an Elasticsearch cluster in the AWS Cloud that has the compute and storage resources that you specify. For example, you can specify the number of instances, instance types, and storage options.

Additionally, Amazon ES offers the following benefits of a managed service:

- Cluster scaling options
- Self-healing clusters
- Replication for high availability
- Data durability
- Enhanced security
- Node monitoring

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

There are no upfront costs to set up clusters, and you pay only for the service resources that you use.

Features of Amazon Elasticsearch Service

Amazon ES provides the following features:

- Multiple configurations of CPU, memory, and storage capacity, known as *instance types*
- Storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple geographical locations for your resources, known as *regions* and *Availability Zones*
- Cluster node allocation across two Availability Zones in the same region, known as *zone awareness*
- Security with AWS Identity and Access Management (IAM) access control
- Dedicated master nodes to improve cluster stability
- Domain snapshots to back up and restore Amazon ES domains and replicate domains across Availability Zones
- Kibana for data visualization
- Integration with Amazon CloudWatch for monitoring Amazon ES domain metrics
- 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

How to Get Started with Amazon Elasticsearch Service

To get started, sign up for an AWS account if you don't already have one. For more information, see Signing Up for AWS (p. 3).

After you are set up, complete the Getting Started (p. 8) tutorial for Amazon Elasticsearch Service. Consult the following introductory topics if you need more information while learning about the service.

Get Up and Running

- Signing Up for AWS (p. 3)
- Accessing Amazon Elasticsearch Service (p. 3)
- Getting Started with Amazon Elasticsearch Service (p. 8)

Basics

- Region and Endpoints for Amazon Elasticsearch Service (p. 3)
- Resource Names and AWS Namespaces
- Scaling (p. 4)
- Versions (p. 6)

Instance Types and Storage

- Choosing an Instance Type
- Configuring EBS-based Storage (p. 23)

Security

- Signing Service Requests (p. 5)

- Creating Access Policies (p. 26)

Signing Up for AWS

If you're not already an AWS customer, create an account. If you already have an AWS account, you are automatically signed up for Amazon ES. Your AWS account enables you to access Amazon ES and other services in the AWS platform, such as Amazon Simple Storage Service (Amazon S3) and Amazon Elastic Compute Cloud (Amazon EC2). There are no sign-up fees, and charges are not incurred until you create a domain. As with other AWS services, you pay only for the resources that you use.

You can use your AWS Identity and Access Management (IAM) user name and password to sign in to the AWS Management Console if you have an account with the IAM service. IAM allows you to securely control access to AWS and resources in your AWS account.

To create an AWS account

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Follow the instructions to sign up. You will need to enter payment information before you can begin using Amazon ES.

Accessing Amazon Elasticsearch Service

You can access Amazon Elasticsearch Service through the Amazon Elasticsearch Service console, the AWS SDKs, or the AWS CLI.

- The Amazon ES console enables you to easily create, configure, and monitor your domains, and also upload data. Using the console is the easiest way to get started with Amazon ES and provides a central command center for ongoing management of your domains.
- The AWS SDKs support all of the Amazon Elasticsearch Service API operations, making it easy to manage and interact with your domains using your preferred technology. The SDKs automatically sign requests as needed using your AWS credentials.
- The AWS CLI wraps all of the Amazon Elasticsearch Service API operations to provide a simple way to create and configure domains. The AWS CLI automatically signs requests as needed using your AWS credentials.

For information about Elasticsearch APIs and features, see the Elasticsearch documentation.

Regions and Endpoints for Amazon Elasticsearch Service

Amazon ES provides regional endpoints for accessing the configuration API and domain-specific endpoints for accessing the search API. You use the configuration service to create and manage your domains. The region-specific configuration service endpoints are of the following form:

`es.region.amazonaws.com`

For example, `es.us-east-1.amazonaws.com`. For a list of supported regions, see Regions and Endpoints in the *AWS General Reference*.

Amazon ES provides a single service endpoint for both search and data services:

- `http://search-domainname-domainid.us-east-1.es.amazonaws.com`

A domain's search endpoint is used to upload data and submit search requests.

Scaling in Amazon Elasticsearch Service

A domain has one or more Elasticsearch instances, each with a finite amount of RAM, CPU, and storage resources for indexing data and processing requests. The number of Elasticsearch instances needed by a domain depends on the documents in your collection and the volume and complexity of your Elasticsearch requests. When you create a domain, you select an initial number of Elasticsearch instances and an instance type. However, these initial choices might not be adequate as the quantity and size of data increase and as Elasticsearch requests increase in number and complexity. You can accommodate the growth by scaling your Amazon ES domain using the guidelines in the following table.

Domain Change	Scaling Guidelines
Increase in data quantity Increase in data size	Use the following guidelines to scale for both increased data quantity and data size: • Choose a larger instance type or add additional instances • Increase the size of the EBS volume
Increase in traffic due to Elasticsearch request volume and complexity	Use the following guidelines to scale for increased traffic: • Choose a larger instance type • Add additional instances • Add replica shards Replica shards provide failover; a replica shard is promoted to a primary shard if a cluster node containing a primary shard fails. For more information about replica shards, see Shards and Replicas in the Elasticsearch documentation.

Scaling for Increased Data

Each Amazon ES domain has one or more search indices. The index stores data in one or more shards distributed across the search instances in your cluster. Amazon ES will automatically migrate shards between search instances as your cluster grows. However, the number of primary shards is fixed when the index is created. The number of primary shards defines the maximum amount of data that can be stored in an index. For more information about indices and shards, see Add an Index in the Elasticsearch documentation.

If you choose to add instances, the index shards are distributed among the available search instances. For more information, see Scale Horizontally in the Elasticsearch documentation. Choosing a larger instance type provides larger local storage for your cluster. Use larger EBS volumes to accommodate larger indices.

Scaling for Increased Traffic

Search and document retrieval requests can be served by primary or replica shards. The more replica shards a cluster has, the more search requests the cluster can handle. Larger instance types have more hardware resources, such as RAM and CPU, which allow each shard to perform better. For more information, see Shards and Replicas in the Elasticsearch documentation.

Signing Service Requests

If you use a programming language for which AWS provides an SDK, we recommend that you use the SDK to submit HTTP requests to AWS. All of the AWS SDKs greatly simplify the process of signing requests, and save you a significant amount of time when compared to natively accessing the Elasticsearch APIs. The SDKs integrate easily with your development environment and provide easy access to related commands. You can also use the Amazon ES console and AWS CLI to submit signed requests with no additional effort.

If you choose to call the Elasticsearch APIs directly, you must sign your own requests. Configuration service requests must always be signed. All requests must be signed unless you configure anonymous access for those services. Use the following procedure to sign a request:

1. Calculate a digital signature using a cryptographic hash function. The input must include the text of your request and your secret access key.

   The function returns a hash value based on your input.
2. Include the digital signature in the Authorization header of your request.

   The service recalculates the signature using the same hash function and input that you used. If the resulting signature matches the signature in the request, the service processes the request. Otherwise, the service rejects the request.

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

Choosing an Instance Type

An instance type defines the memory, CPU, storage capacity, and hourly cost for an *instance*, the Amazon Machine Image (AMI) that runs as a virtual server in the AWS Cloud. You should choose the instance type and the number of instances based on the anticipated size of the Elasticsearch indices, shards, and replicas that you intend to create on your cluster. Amazon ES supports the following instance types:

- t2.micro.elasticsearch
- t2.small.elasticsearch
- t2.medium.elasticsearch
- m3.medium.elasticsearch
- m3.large.elasticsearch
- m3.xlarge.elasticsearch
- m3.2xlarge.elasticsearch
- r3.large.elasticsearch
- r3.xlarge.elasticsearch
- r3.2xlarge.elasticsearch
- r3.4xlarge.elasticsearch
- r3.8xlarge.elasticsearch
- i2.xlarge.elasticsearch
- i2.2xlarge.elasticsearch

Note

- The sa-east-1 and us-east-2 regions do not support r3 and i2 instance types.

- The us-east-2, ap-northeast-2, and ap-south-1 regions support m4 instance types instead of m3 instance types.

For more information, see Instance Types in the Amazon EC2 documentation.

Using Amazon EBS Volumes for Storage

You have the option of configuring your Amazon ES domain to use an Amazon EBS volume for storing indices rather than the default storage provided by the instance. An Amazon EBS volume is a durable, block-level storage device that you can attach to a single instance. Amazon ES supports the following EBS volume types:

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

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

Choosing an Elasticsearch Version

Amazon ES currently supports two Elasticsearch versions: 1.5 and 2.3. Compared to Elasticsearch version 1.5, version 2.3 offers improved performance, memory management, and security. It also offers several additional features, including the following:

- Pipeline aggregations to perform advanced analytics, such as moving averages and derivatives
- Enhancements to geospatial queries
- Configurable store compression

For more information about the differences between Elasticsearch 1.5 and 2.3, see the Elasticsearch documentation. For information about the Elasticsearch APIs that Amazon ES supports for both 1.5 and 2.3, see Supported Elasticsearch Operations (p. 67).

If you are starting a new Elasticsearch project, we strongly recommend that you choose version 2.3. If you have an existing 1.5 domain, you can choose to keep the domain or migrate your data to a new 2.3 domain. For more information, see Migrating Data in a Domain from Elasticsearch Version 1.5 to Version 2.3 (p. 65).

Related Services

Amazon ES is commonly used with the following services:

AWS CloudTrail
: Use AWS CloudTrail to get a history of the Amazon ES API calls and related events for your account. CloudTrail is a web service that records API calls from your accounts and delivers the resulting log files to your Amazon S3 bucket. You can also use CloudTrail to track changes that were made to your AWS resources. For more information, see Auditing Amazon Elasticsearch Service Domains with AWS CloudTrail (p. 49).

Amazon CloudWatch
: An Amazon ES domain automatically sends metrics to Amazon CloudWatch so that you can gather and analyze performance statistics. You can monitor these metrics by using the AWS CLI or the AWS SDKs. For more information, see Monitoring Cluster Metrics and Statistics with Amazon CloudWatch (Console) (p. 42).

Amazon Kinesis
: Amazon Kinesis is a managed service that scales elastically for real-time processing of streaming data at a massive scale. Amazon ES provides Lambda sample code for integration with Amazon Kinesis. For more information, see Streaming Data to Amazon ES From Amazon Kinesis (p. 61).

Amazon S3
: Amazon Simple Storage Service (Amazon S3) is storage for the Internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. Amazon ES provides Lambda sample code for integration with S3. For more information, see Streaming Data to Amazon ES From Amazon S3 (p. 58).

AWS IAM
: AWS Identity and Access Management (IAM) is a web service that you can use to manage users and user permissions in AWS. Use IAM to create user-based access policies for your Amazon ES domains. See the IAM documentation for more information about using IAM to create user polices.

Amazon ES integrates with the following services to provide data ingestion:

AWS Lambda
: AWS Lambda is a zero-administration compute platform for back-end web developers that runs your code in the AWS Cloud and provides you with a fine-grained pricing structure. Amazon ES provides sample code to run on Lambda that integrates with Amazon Kinesis and Amazon S3. For more information, see Streaming Data to Amazon ES (p. 57).

Amazon DynamoDB
: Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. Amazon ES provides a Logstash plugin to support DynamoDB Streams and sign AWS service requests.

Pricing for Amazon Elasticsearch Service

With Amazon Web Services, you pay only for what you use. For Amazon ES, you pay for each hour of use of an EC2 instance. You also have the option of paying for additional storage based on the cumulative size of EBS volumes attached to the data nodes in your domain.

If you qualify for the AWS Free Tier, you receive up to 750 hours per month of use with the t2.micro.elasticsearch instance type, as well as up to 10 GB of magnetic or general provisioned EBS storage. For more information, see AWS Free Tier.

Getting Started with Amazon Elasticsearch Service Domains

This tutorial shows you how to use Amazon Elasticsearch Service (Amazon ES) to create and configure a domain. 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 analytics. With Amazon ES, you get direct access to Elasticsearch APIs so that existing code and applications work seamlessly together. Currently, Amazon ES supports Elasticsearch versions 1.5 and 2.3.

To get started using the service, you create an Amazon ES domain. An Amazon ES domain is an Elasticsearch cluster in the AWS Cloud that has the compute and storage resources that you specify. For example, you can specify the number of instances, instance types, and storage options. Amazon ES includes data connectors for ingesting data into Amazon ES domains and analytics tools to search for patterns in your indexed data.

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

1. Signing up for AWS (p. 9)
2. Creating an Amazon ES domain (p. 9)
3. Configuring an access policy for an Amazon ES domain (p. 11)
4. Uploading data to an Amazon ES domain for indexing (p. 12)
5. Searching documents from an Amazon ES domain (p. 13)
6. Deleting an Amazon ES domain (p. 13)

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

> **Note**
> If you want to use the AWS CLI to complete the tutorial, you must use version 1.8.9 or later. To see which version of the AWS CLI you have installed, run the `aws` command with the `--version` option: `aws --version`.

Step 1: Signing Up for AWS

If you're not already an AWS customer, create an account. If you already have an AWS account, you are automatically signed up for Amazon Elasticsearch Service (Amazon ES). Your AWS account enables you to access Amazon ES and other AWS services, such as Amazon Simple Storage Service (Amazon S3) and Amazon Elastic Compute Cloud (Amazon EC2). There are no sign-up fees, and charges are not incurred until you create a domain. You pay only for the resources that you use.

For console access, use your IAM user name and password to sign in to the AWS Management Console using the IAM sign-in page. IAM lets you securely control access to AWS services and resources in your AWS account. For more information about creating access keys, see How Do I Get Security Credentials? in the *AWS General Reference*.

To create an AWS account

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. Under **Analytics**, choose **Elasticsearch Service**.
3. Follow the instructions to sign up. You need to enter payment information before you can begin using Amazon ES.

Step 2: Creating an Amazon ES Domain

An Amazon Elasticsearch Service (Amazon ES) domain encapsulates the Elasticsearch engine instances that process HTTP requests to AWS, the indexed data that you want to search, snapshots of the domain, access policies, and metadata. You can create an Amazon ES domain by using the Amazon ES console, the AWS CLI, or the AWS SDK.

You need only one piece of information to create an Amazon ES domain with the Amazon ES console: a domain name. 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 (-)

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 **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 version 2.3. For more information, see Choosing an Elasticsearch Version (p. 6).
5. Choose **Next**.
6. For **Instance count**, choose the number of instances that you want.
7. For **Instance type**, choose an instance type for the Amazon ES domain.
8. If you want a dedicated master node, select **Enable dedicated master**. Master nodes improve cluster stability by separating cluster management tasks, performed only on the dedicated master node, from index and search requests. For more information, see Using Dedicated Master Nodes with an Amazon ES Domain (p. 34).

9. If you want to enable zone awareness, select **Enable zone awareness**. Zone Awareness allocates nodes across two Availability Zones in the same region. Zone awareness requires an even number of instances in the instance count. For more information, see About Zone Awareness (p. 36).
10. If you want to use an EBS volume storage, for **Storage type**, choose **EBS**.

    The **EBS volume type** and **EBS volume size** fields appear.

    a. For **EBS volume type**, choose the external storage type. 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.

       Calculate the total amount of EBS-based storage for the Amazon ES domain using the following formula: (number of data nodes) * (EBS volume size).
11. For **Automated snapshot start hour**, choose the hour of the day when automated snapshots will be taken.
12. (Optional) Choose **Advanced options**.

    a. (Optional) If you want to configure access to domain sub-resources, for **rest.action.multi.allow_explicit_index**, choose **false**.

       Disabling this property prevents users from bypassing access control for sub-resources. For more information about access control, see URL-based Access Control in the Elasticsearch documentation. For more information about access policies for sub-resources, see Configuring Access Policies (p. 26).
    b. (Optional) For **indices.fielddata.cache.size**, specify the percentage of heap space to allocate to the field data cache.

       By default, this setting is unbounded. For more information about the field data cache, see Field Data in the Elasticsearch documentation.
13. Choose **Next**.
14. Enter an access policy for the domain or select one of the policy templates from **Select a template**, and then choose **Next**.
15. Review the new domain configuration, and then choose **Confirm and create**.
16. Choose **OK**.

To create an Amazon ES domain (AWS CLI)

- Run one of the following commands to create an Amazon ES domain.

  The first command creates a domain named *movies* with the default Elasticsearch version 1.5 with one instance of the default m3.medium.elasticsearch instance type. The domain also uses the default storage provided by the search engine instance:

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

  The following command creates an Amazon ES domain named *movies* with Elasticsearch version 2.3 with two instances of the m3.medium.elasticsearch instance type. The domain uses a 100 GB Magnetic disk EBS volume as storage for each data node:

```
aws es create-elasticsearch-domain --domain-name movies --
elasticsearch-version 2.3 --elasticsearch-cluster-config
 InstanceType=m3.medium.elasticsearch,InstanceCount=2 --ebs-options
 EBSEnabled=true,VolumeType=standard,VolumeSize=100
```

The following command creates an Amazon ES domain named *movies* with Elasticsearch version 2.3 with five instances of the m3.large.elasticsearch instance type. The domain uses a 100 GB General Purpose (SSD) EBS volume as storage for each data node:

```
aws es create-elasticsearch-domain --domain-name movies --
elasticsearch-version 2.3 --elasticsearch-cluster-config
 InstanceType=m3.large.elasticsearch,InstanceCount=5 --ebs-options
 EBSEnabled=true,VolumeType=gp2,VolumeSize=100
```

The following command creates an Amazon ES domain named *movies* with the default Elasticsearch version 1.5 with ten instances of the m3.xlarge.elasticsearch instance type. The domain uses a 100 GB Provisioned IOPS (SSD) EBS volume as storage for each data node and an IOPS value of 1000:

```
aws es create-elasticsearch-domain --domain-name movies --elasticsearch-
cluster-config InstanceType=m3.xlarge.elasticsearch,InstanceCount=10 --
ebs-options EBSEnabled=true,VolumeType=io1,VolumeSize=100,Iops=1000
```

Note
Initializing a domain and its resources takes approximately ten minutes. When initialization is complete, the endpoint of the domain is available for index and Amazon ES requests.

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. 77), including the `describe-elasticsearch-domain` command. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Step 3: Configuring an Access Policy for an Amazon ES Domain

Amazon ES domains currently support resource-based, IP-based, and IAM user and role-based access policies. Resource-based IAM access policies enable you to configure anonymous access to an Amazon ES domain from select IP addresses or IP address ranges.

If you created your Amazon ES domain by using the console procedure that is described in step 2, you have already assigned an access policy. However, it is not uncommon to modify an access policy after creating a domain.

Although you can configure access policies with the Amazon ES console, the AWS CLI, or the AWS SDKs, the console provides four preconfigured policies, as well as a multi-line section for entering custom policies.

To configure an access policy (console)

1. Go to https://aws.amazon.com, and then choose **Sign In to the Console**.
2. In the navigation pane, under **My domains**, choose the domain that you want to configure.
3. Choose **Modify access policy**.

4. Edit your current access policy or select one of the policy templates from **Select a template**, and then choose **Submit**. For more information, see Configuring Access for Amazon ES Domains (p. 26).

   The status of your domain changes from **Active** to **Processing**. The status must return to **Active** before your modified access policy takes effect.

To configure an access policy (AWS CLI)

- Run the following command to authorize access to the new domain for the *movies* domain only from the specified IP addresses:

```
aws es update-elasticsearch-domain-config --endpoint https://
es.us-west-1.amazonaws.com --domain-name movies --access-
policies '{"Version": "2012-10-17", "Statement": [{"Action":
 "es:ESHttp*","Principal":"*","Effect": "Allow", "Condition":
 {"IpAddress":{"aws:SourceIp":["192.0.2.0/32"]}}}]}"
```

To configure an access policy (AWS SDKs)

The AWS SDKs (except the Android and iOS SDKs) support all the actions defined in the Amazon Elasticsearch Service Configuration API Reference (p. 77), including the `update-elasticsearch-domain-config` command. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Step 4: Uploading Data to an Amazon ES Domain for Indexing

You can upload data to an Amazon ES domain for indexing by 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 described in the same JSON file. Each line in the JSON file must end with a newline character, `\n`, including the last line.

> **Note**
> Standard clients, such as curl, cannot perform the request signing that is required of identity-based access policies. You must use an IP address-based access policy that allows anonymous access to successfully perform the instructions for this step.

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 search-movies-4f3nw7eiia2xiynjr55a2nao2y.us-
west-1.es.amazonaws.com/movies/movie/tt0116996 -d '{"directors" : ["Tim
 Burton"],"genres" : ["Comedy","Sci-Fi"],"plot" : "The Earth is invaded by
 Martians with irresistible weapons and a cruel sense of humor.","title" :
 "Mars Attacks!","actors" : ["Jack Nicholson","Pierce Brosnan","Sarah
 Jessica Parker"],"year" : 1996}'
```

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

- Run the following command to upload multiple documents to the *movies* domain:

```
curl -XPOST 'http://search-movies-4f3nw7eiia2xiynjr55a2nao2y.us-
west-1.es.amazonaws.com/_bulk' --data-binary @bulk_movies.json
```

Tip
The service supports migrating data from manual snapshots taken on both Amazon ES and self-managed Elasticsearch clusters. Restoring indices from a self-managed Elasticsearch cluster is a common way to migrate data into Amazon ES.

Step 5: Searching Documents in an Amazon ES Domain

To search documents in an Amazon ES domain, use the Elasticsearch search API from the command line. Alternatively, use the Kibana dashboard to search documents in the domain. You can find the endpoint to the Kibana dashboard for your domain in the Amazon ES console. The endpoint matches the domain endpoint, but with `/_plugin/kibana/` appended.

Note
Standard clients, such as curl, cannot perform the request signing that is required of identity-based access policies. You must use an IP address-based access policy that allows anonymous access to successfully perform the instructions for this step.

To search documents (AWS CLI)

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

```
curl -XGET 'search-movies-4f3nw7eiia2xiynjr55a2nao2y.us-
west-1.es.amazonaws.com/movies/_search?q=nightmare'
```

To search documents from an Amazon ES domain by using Kibana (console)

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.
2. For **Search**, type your query, and then choose **Enter**.

Step 6: Deleting an Amazon ES Domain

This step shows how to delete an Amazon ES domain. If you created a test domain during this Getting Started tutorial, such as a *movies* domain, delete the domain when you are finished experimenting with it or you might incur 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 <domain_name>** 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 --endpoint https://es.us-
west-1.amazonaws.com --domain-name movies
```

Note
Deleting a domain deletes all billable Amazon ES resources. However, manual snapshots of the domain created with the native Elasticsearch API are not deleted. Consider saving a snapshot if you might need to recreate the Amazon ES domain in the future. If you do not 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. 77), including the `delete-elasticsearch-domain` command. 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 the hardware, software, and data exposed by Amazon Elasticsearch Service endpoints.

Unlike the brief instructions in the Getting Started (p. 8) 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 the AWS SDKs.

- Creating Amazon ES Domains (p. 15)
- Configuring Amazon ES Domains (p. 20)
- Configuring EBS-based Storage (p. 23)
- Configuring Access Policies (p. 26)
- Configuring Snapshots (p. 31)
- Configuring Advanced Options (p. 32)

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.
5. For **Version**, choose an Elasticsearch version for your domain. We recommend that you choose version 2.3. For more information, see Choosing an Elasticsearch Version (p. 6).
6. Choose **Next**.
7. For **Instance count**, choose the number of instances that you want.

   You can choose up to twenty instances for each Amazon ES domain. We recommend that you choose more than two instances to avoid potential Elasticsearch issues, such as the split brain issue. We also recommend that you have a replica for each index to avoid potential data loss. For more information about replicas, see Shards and Replicas in the Elasticsearch documentation.

   Note
   You can use a T2 instance type only if the instance count for the domain is 10 or fewer.
8. For **Instance type**, choose an instance type for the data nodes.

   For a list of supported Amazon ES instance types, see the table that describes the `--elasticsearch-cluster-configure` option (p. 22) for the AWS CLI.

   Note

   - The sa-east-1 and us-east-2 regions do not support r3 and i2 instance types.
   - The us-east-2, ap-northeast-2, and ap-south-1 regions support m4 instance types instead of m3 instance types.
9. If you need to 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 Using Dedicated Master Nodes (p. 34).

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

      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.
      - You can use a T2 instance type only if the instance count is 10 or fewer.
   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) To provide high availability for data nodes, 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 also must use the Amazon ES API to replicate your data for your Amazon ES cluster. This allows for the even distribution of shards across two Availability Zones. For more information, see Using Zone Awareness for High Availability (p. 36).
11. For **Storage type**, choose either **Instance** (the default) or **EBS**.

Use an EBS volume for storage rather than the storage attached to the selected instance type if your Amazon ES domain requires more storage. Domains with very large indices or large numbers of indices often benefit from the increased storage capacity of EBS volumes. If you choose **EBS**, the following fields 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.

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

   Calculate the total amount of EBS-based storage for the Amazon ES domain using the following formula: (number of data nodes) * (EBS volume size). For more information about supported EBS volume types and sizes, see Configuring EBS-based Storage (p. 23). 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. For specific limits, see the EBS Resource (p. 107) table in this guide.

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

    By default, the service takes an automated snapshot within an hour of midnight.

13. (Optional) Choose **Advanced options**.

    a. (Optional) If you want to configure access to domain sub-resources, for **rest.action.multi.allow_explicit_index**, choose **false**.

       Disabling this property prevents users from bypassing access control for sub-resources. For more information about access control, see URL-based Access Control in the Elasticsearch documentation. For more information about access policies for sub-resources, see Configuring Access Policies (p. 26).

    b. (Optional) For **indices.fielddata.cache.size**, specify the percentage of heap space to allocate to the field data cache.

       By default, this setting is unbounded. For more information about the field data cache, see Field Data in the Elasticsearch documentation.

       Tip
       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 of these use cases. However, if you have very large indices, you might need a larger field data cache.

14. Choose **Next**.
15. On the **Set up access policy** page, select a pre-configured 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 **Edit the access policy** box.

    Amazon Elasticsearch Service offers several ways to configure access to your Amazon ES domains. The console provides preconfigured access policies that you can customize to the specific needs of your domain, as well as the ability to import access policies from other Amazon ES domains. The service also allows you to specify separate, fine-grained access policies to each domain sub-resource. For example, you can assign a different policy to each index in your Amazon ES domain. For more information, see Configuring Access Policies (p. 26).

16. Choose **Next**.
17. On the **Review** page, review your domain configuration, and then choose **Confirm and create**.
18. 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.

Syntax

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

[--elasticsearch-version <value>]

[--elasticsearch-cluster-config <value>]

[--ebs-options <value>]

[--access-policies <value>]

[--snapshot-options <value>]

[--advanced-options <value>]

[--cli-input-json <value>]

[--generate-cli-skeleton]
```

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

Optional Parameters

Parameter	Description
`--elasticsearch-version`	Specifies the Elasticsearch version of the domain. Currently, Amazon ES supports versions 1.5 and 2.3. If not specified, the default value is 1.5. For more information, see Choosing an Elasticsearch Version (p. 6).
`--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 an Amazon ES Cluster (p. 22).
`--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 Options (p. 24).
`--access-policies`	Specifies the access policy for the domain. For more information, see Configuring Access Policies (p. 28).
`--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

Parameter	Description
	snapshot is taken anytime between midnight and 1:00 AM. For more information, see Configuring Automated Index Snapshots (p. 31).
`--advanced-options`	Specifies whether to allow references to indices in the bodies of HTTP request objects. For more information, see Configuring Advanced Options (p. 33).
`--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*.

Examples

The first example demonstrates the following Amazon ES domain configuration:

- Creates an Amazon ES domain named *weblogs* with Elasticsearch version 2.3
- Populates the domain with two instances of the m3.medium.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 2.3 --elasticsearch-cluster-config
 InstanceType=m3.medium.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 2.3
- Populates the domain with six instances of the m3.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 2.3 --elasticsearch-cluster-config
 InstanceType=m3.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 2.3
- Populates the domain with ten instances of the m3.xlarge.elasticsearch instance type
- Populates the domain with three instances of the m3.medium.elasticsearch instance type to serve as a dedicated master node
- 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 sub-resource, the `_search` API
- Configures automated daily snapshots of the indices for 03:00 UTC

```
aws es create-elasticsearch-domain --domain-name weblogs --
elasticsearch-version 2.3 --elasticsearch-cluster-config
 InstanceType=m3.xlarge.elasticsearch,InstanceCount=10,DedicatedMasterEnabled=true,Dedicate
 --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 successfully create an Amazon ES domain, the CLI ignores attempts to create another domain with same name. The CLI does not report an error.

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. 77), including `create-elasticsearch-domain`. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring Amazon ES Domains

Update your Amazon ES domain configuration with any of the following changes to meet the demands of increased traffic and data:

- Change the instance count
- Change the instance type
- Enable or disable a dedicated master node
- Enable or disable zone awareness
- Configure EBS-based storage
- Change the start time for automated snapshots of domain indices
- Configure a native Amazon ES property

Note
For information about configuring a domain to use an EBS volume for storage, see Configuring EBS-based Storage (p. 23).

Configuring Amazon ES Domains (Console)

Use the following procedure to make updates to 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.

      Note

      - The sa-east-1 and us-east-2 regions do not support r3 and i2 instance types.
      - The us-east-2, ap-northeast-2, and ap-south-1 regions support m4 instance types instead of m3 instance types.

   b. If you want to change the instance count, for **Instance count**, choose an integer from one to twenty.

   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 Amazon ES cluster. For more information, see Using Dedicated Master Nodes (p. 34).

      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.

      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.

   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. Choose **Advanced options**.

      i. (Optional) If you want to configure access to domain sub-resources, for **rest.action.multi.allow_explicit_index**, choose **false**.

      Disabling this property prevents users from bypassing access control for sub-resources. For more information about access control, see URL-based Access Control in the Elasticsearch documentation. For more information about access policies for sub-resources, see Configuring Access Policies (p. 26).

      ii. For `rest.action.multi.allow_explicit_index`, choose **false**.

iii. (Optional) For **indices.fielddata.cache.size**, specify the percentage of heap space to allocate to the field data cache.

By default, this setting is unbounded. For more information about the field data cache, see Field Data in the Elasticsearch documentation.

Tip
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 of these cases. However, if you have very large indices you might need a larger field data cache.

g. 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>,DedicatedMasterT
```

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: • t2.micro.elasticsearch • t2.small.elasticsearch • t2.medium.elasticsearch • m3.medium.elasticsearch • m3.large.elasticsearch • m3.xlarge.elasticsearch • m3.2xlarge.elasticsearch • r3.large.elasticsearch • r3.xlarge.elasticsearch • r3.2xlarge.elasticsearch • r3.4xlarge.elasticsearch • r3.8xlarge.elasticsearch • i2.xlarge.elasticsearch • i2.2xlarge.elasticsearch	The hardware configuration of the computer that will host the instance. The default is m3.medium.elasticsearch.
`InstanceCount`	Integer	The number of instances in the Amazon ES domain. The default is one, and the maximum is twenty.

Parameter	Valid Values	Description
`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 will host the master node. The default is m3.medium.elasticsearch.
`DedicatedMasterCount`	Integer	The number of instances used for the dedicated master node. The default is three.
`ZoneAwarenessEnabled`	`true` or `false`	Specifies whether to enable zone awareness for the Amazon ES domain. The default value is `false`.

Note

- The sa-east-1 and us-east-2 regions do not support r3 and i2 instance types.
- The us-east-2, ap-northeast-2, and ap-south-1 regions support m4 instance types instead of m3 instance types.

Examples

The following example creates an Amazon ES domain named `mylogs` with Elasticsearch version 2.3 with two instances of the m3.medium.elasticsearch instance type and zone awareness enabled:

```
aws es create-elasticsearch-domain --domain-name mylogs --
elasticsearch-version 2.3 --elasticsearch-cluster-config
 InstanceType=m3.medium.elasticsearch,InstanceCount=2,DedicatedMasterEnabled=false,ZoneAwar
```

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=m3.large.elasticsearch,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedM
```

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. 77), including `update-elasticsearch-domain-config`. 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 very large data sets, 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`.

Caution
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.

   Calculate the total amount of EBS-based storage for the Amazon ES domain using the following formula: (number of data nodes) * (EBS volume size). See Amazon Elasticsearch Service Limits (p. 107) for the minimum and maximum size of EBS volumes.
8. Choose **Submit**.

Tip
You must 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. For information about the maximum size of supported EBS volumes in an Amazon ES domain, see Understanding Amazon Elasticsearch Service Limits (p. 107).

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 new Amazon ES domain.

`VolumeSize`	Integer	Specifies the size of the EBS volume for each data node. 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.
		
		Minimum and maximum EBS volume sizes by instance type:
		
		`t2.micro.elasticsearch`, `t2.small.elasticsearch`, `t2.medium.elasticsearch`
		Minimum: 10 GB; Maximum: 35 GB for each data node
		All other small and medium instance types
		Minimum: 10 GB; Maximum: 100 GB
		All Large, XLarge, and 2XLarge instance types
		Minimum: 10 GB; Maximum: 512 GB
		
		Minimum and maximum EBS volume sizes by volume type:
		
		`gp2`
		Minimum: 10 GB; Maximum: 512 GB for each data node
		`io1`
		Minimum: 35 GB; Maximum: 512 GB for each data node
		`standard`
		Minimum: 10 GB; Maximum: 100 GB for each data node
		
		For example, the maximum size for a `gp2` EBS volume type attached to a `t2.small.elasticsearch` instance type is 35 GB, not 512 GB, because the maximum EBS volume size for this instance type is 35 GB. If you want a larger `gp2` EBS volume, you must attach it to a larger instance type. Calculate the total amount of EBS-based storage for the Amazon ES

Parameter	Valid Values	Description
		domain using the following formula: (number of data nodes) * (EBS volume size).
`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, and the maximum value is 4000.

Tip
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 types, see Amazon EBS Volume Types in the Amazon EC2 documentation.

Examples

The following example creates a domain named `mylogs` with Elasticsearch version 2.3 with a 10 GB general purpose EBS volume:

```
aws es create-elasticsearch-domain --domain-name=mylogs --elasticsearch-
version 2.3 --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. 77), including the `--ebs-options` parameter to the `update-elasticsearch-domain-config` command. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring Access Policies

Amazon Elasticsearch Service offers several ways to configure access to your Amazon ES domains. The console provides preconfigured access policies that you can customize to the specific needs of your domain as well as the ability to import access policies from other Amazon ES domains. The service also allows you to specify separate, fine-grained access policies to each domain sub-resource. For example, you can assign a different policy to each index in your Amazon ES domain.

Access Configuration Method	Description
Resource-based access policy	Resource-based access policies are attached to a specific Amazon ES domain. A resource-based policy specifies who can access the

Access Configuration Method	Description
	endpoint of the domain. Use the `Principal` policy element to specify who is allowed access. Use the `Resource` policy element to specify which resources are accessible.
IP-based policy	IP-based access policies restrict access to an Amazon ES domain to one or more specific IP addresses. IP-based policies can also be configured to allow anonymous access, which enables you to submit unsigned requests to an Amazon ES domain. Use the `Condition` policy element to specify which IP addresses are allowed to access the service.
IAM user and role-based access policies	Amazon ES also supports access policies based on IAM users and roles. Use the IAM service to specify which users and roles can access the service and what sub-resources they can use.

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	This policy is used to allow or deny access to one or more AWS accounts or IAM users.
Allow open access to the domain	This policy is not recommended because it allows anyone to upload documents to the domain. The policy is intended only as a convenience for testing and debugging.
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.
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.
Copy access policy from another domain	This policy provides a convenient way to import an existing access policy from another domain.

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 new Amazon ES domain.

Amazon Elasticsearch Service supports all of the policy elements documented in the IAM Policy Elements Reference, including the most common elements described in the following table.

Valid Values	JSON Policy Element	Description
The current version of the policy language is `2012-10-17`. All access policies should specify this value.	`Version`	Specifies the language version for the access policy.
`Allow` or `Deny`	`Effect`	Specifies whether the statement allows or blocks access to the specified actions.
Any string	`Sid`	A descriptive name for the policy statement. This field is optional.
Amazon ES supports the following actions for HTTP methods. You can attach a separate access policy to each HTTP method: • `es:ESHttpDelete` • `es:ESHttpGet` • `es:ESHttpHead` • `es:ESHttpPost` • `es:ESHttpPut` Amazon ES also supports the following actions for the service configuration APIs: • `es:CreateElasticsearchDomain` • `es:DescribeElasticsearchDomain` • `es:DescribeElasticsearchDomains` • `es:DescribeElasticsearchDomainConfig`	`Action`	Specifies the Amazon ES actions to which the access policy applies. Assign a value of `"Action":"es:*"` to allow full access to the domain endpoint with any HTTP method.

Valid Values	JSON Policy Element	Description
• `es:DeleteElasticsearchDomain` • `es:ListDomainNames` • `es:AddTags` • `es:ListTags` • `es:RemoveTags` • `es:UpdateElasticsearchDomainConfig` For a description of each API, including the HTTP request method required for each, see the Amazon ES Configuration API Reference (p. 77).		

Use the following syntax to specify domain resources for Amazon ES:
 	
`arn:aws:es:<region>:<aws_account_id>:domain:<domain-name>/<sub-resource>`
 	
Specify the wildcard (*) as a sub-resource to allow or deny access to all sub-resources:
 	
`arn:aws:es:<region>:<aws_account_id>:domain:<domain-name>/*`
 	
Use the following syntax to allow or deny all access to a specific sub-resource:
 	
`arn:aws:es:<region>:<aws_account_id>:domain:<domain-name>/<sub-resource>/*`
 	
Amazon ES allows you to define a different access policy for each sub-resource, such as indices. You can also define a different access policy for each Amazon ES API. For example, you can limit the scope of the grant permission to only the Amazon ES `_search` API.
 	
`arn:aws:es:<us-east-1:<account-id>:domain/weblogs/_search`
	`Resource`	Specifies the specific object or objects to which the access policy applies.

Amazon ES supports all the conditions described in Available Keys for Conditions in the *Using IAM* guide.	`Condition`	Specifies conditions that determine when the access policy is in effect. When configuring anonymous, IP-based access, specify the IP addresses for which the access rule applies. For example: `"IpAddress": {"aws:SourceIp": ["192.0.2.0/32"]}`.

Valid Values	JSON Policy Element	Description

Any of the following:
• Other AWS accounts
`"Principal":{"AWS": ["arn:aws:iam::<aws_account_id>:root"]}`
• IAM users
`"Principal":{"AWS": [arn:aws:iam::<aws_account_id>:user/<username>}`
	`Principal`	Specifies the AWS account or IAM user that is allowed or denied access to a resource. Specifying a wildcard (*) enables anonymous access to the domain, which is not recommended. If you do enable anonymous access, we strongly recommend that you add an IP-based condition to restrict which IP addresses can submit requests to the Amazon ES domain.
		
		For more information, see Principal in the *IAM Policy Elements Reference*.

Resource-based Policy Example

The following example of a resource-based access policy 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 is granted access to the `index1` domain sub-resource, but will not be able to 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/*" } ] }'
```

IP-based Policy Example

The following example of an IP-based access policy allows anonymous access, but restricts that access to a single range of IP addresses:

```
aws es update-elasticsearch-domain-config --domain-name mylogs
 --access-policies '{"Version": "2012-10-17","Statement":
 [{"Action":"es:*","Principal":"*","Effect":"Allow","Condition":
 {"IpAddress":{"aws:SourceIp":["192.0.2.0/32"]}}}]}'
```

IAM-based Policy Example

You create IAM-based access policies by using the AWS IAM console rather than the Amazon ES console. For information about creating IAM-based access policies, see the IAM documentation.

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. 77), including the `--access-policies` parameter to the `update-elasticsearch-domain-config` command. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Configuring 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 can choose a different time.

> **Caution**
> The service stops taking snapshots of Amazon ES indices while the health of a cluster is RED. Subsequent uploads to indices in a RED cluster, even indices with a health status of GREEN, could be lost in the event of a cluster failure due to the cessation of snapshots. To prevent loss of data, return the health of your cluster to GREEN before uploading additional data to any index in the cluster.

Configuring Snapshots (Console)

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

To configure automatic snapshots

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 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-1 --snapshot-options AutomatedSnapshotStartHour=1
```

Configuring Snapshots (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. 77), including the `--snapshots-options` parameter to the `update-elasticsearch-domain-config` command. 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
: 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 information, see URL-based Access Control. For more information about access policies for sub-resources, see Configuring Access Policies (p. 26).

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

Configuring Advanced Options (Console)

Use the following procedure to disable processing of HTTP requests with explicit references to indices in the request body by using the console. By default, the value is `true`.

To configure advanced options

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. For `rest.action.multi.allow_explicit_index`, choose **false** to disable explicit references to indices in the bodies of HTTP requests.
7. For `indices.fielddata.cache.size`, enter the percentage of Java heap space that is allocated to field data cache.

   By default, this setting is unbounded. For more information about the field data cache, see Field Data in the Elasticsearch documentation.

   Tip
   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 larger field data cache.

8. 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=<true|false>,
indices.fielddata.cache.size=<percentage_heap>
```

Parameter	Valid Values	Description
`--advanced-options`	`rest.action.multi.allow_explicit=<true|false>,`	Specifies whether explicit references to indices are allowed in an HTTP request body. Such references are always allowed in the HTTP request URL. Must be `false` when configuring access to individual sub-resources. By default, the value is `true`.
	`indices.fielddata.cache.size=<percentage_heap>`	Specifies the percentage of heap space to allocate to the field data cache. By default, this setting is unbounded.

Tip
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.

Example

The following example disables explicit references to indices in the HTTP request bodies and 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. 77), including the `--advanced-options` parameter to the `update-elasticsearch-domain-config` command. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

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. You will need to monitor domain metrics to know when to reconfigure your domain. You also have the option of managing your own index snapshots, auditing data-related API calls to your domain, and assigning tags to your domain. You can load data in bulk to your domain using the Logstash plugin provided by the service. This chapter describes how to perform these and other tasks related to managing your domains.

- About Dedicated Master Nodes (p. 34)
- About Zone Awareness (p. 36)
- Working with Manual Index Snapshots (p. 37)
- Monitoring Cluster Metrics and Statistics with Amazon CloudWatch (p. 42)
- Auditing Domains with AWS CloudTrail (p. 49)
- Visualizing Data with Kibana (p. 51)
- Loading Bulk Data with the Logstash Plugin (p. 51)
- Signing Amazon Elasticsearch Service Requests (p. 53)
- Tagging Amazon Elasticsearch Service Domains (p. 54)

About Dedicated Master Nodes

Amazon Elasticsearch Service (Amazon ES) uses dedicated master nodes to increase cluster stability. A dedicated master node is a cluster node that 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 Elasticsearch clusters.

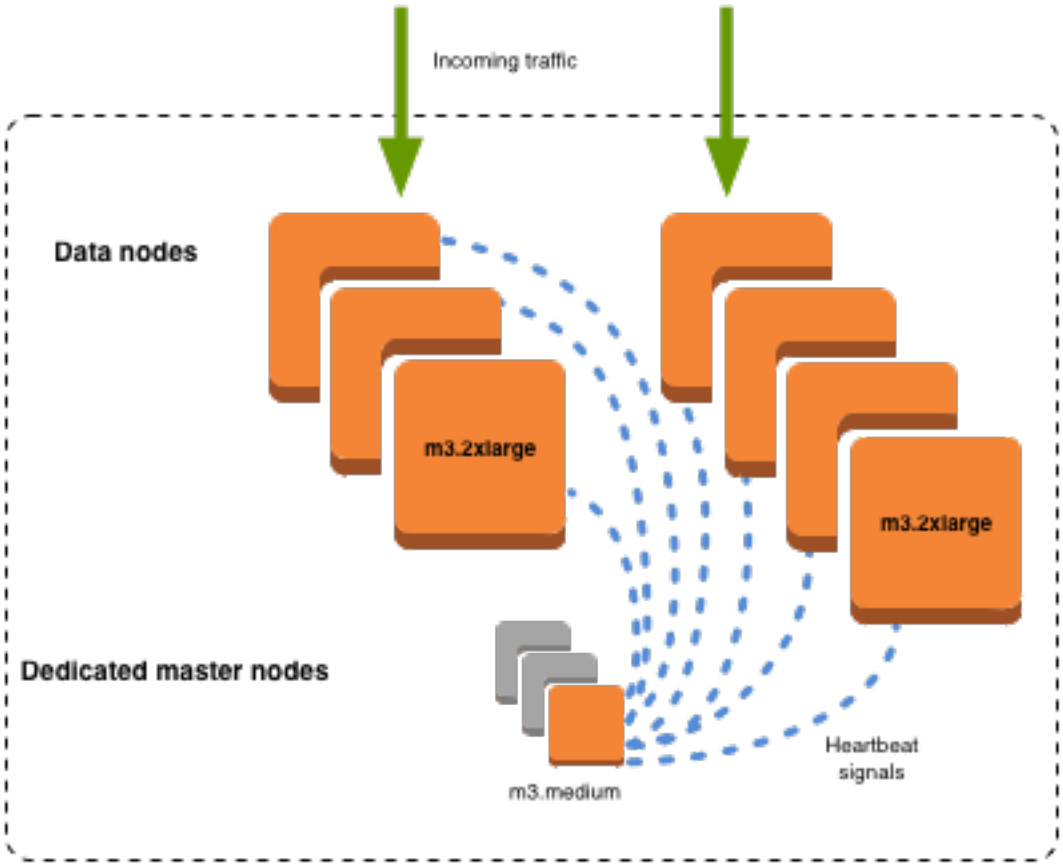
> **Tip**
> We recommend that you allocate three dedicated master nodes for all Amazon ES domains in production.

A dedicated master node performs the following cluster management tasks:

- Tracks all nodes in the cluster
- Tracks the number of indices in the cluster
- Tracks the number of shards belonging to each index

- Maintains routing information for nodes in the cluster
- Updates the cluster state after state changes, such as index creation and the addition and removal of nodes in the cluster
- Replicates changes to the cluster state across all nodes in the cluster
- Monitors the health of all cluster nodes by sending *heartbeat signals*, or 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 grey 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.



Because a dedicated master node does not process search and query requests, the instance type chosen for this role typically does not require a large CPU and memory. Initially, use the m3.medium.elasticsearch instance type and adjust as necessary. Small Elasticsearch clusters with a single index and few index shards will not benefit from a dedicated master node. We recommend that you avoid allocating dedicated master nodes for all small and short-lived Amazon ES domains.

To prevent overloading a dedicated master node, you can monitor usage with the Amazon CloudWatch metrics that are shown in the following table. Use a larger instance type for dedicated master nodes when these metrics reach their respective maximum values.

CloudWatch Metric	Guideline
MasterCPUUtilization	Measures the percentage utilization of the CPU for the dedicated master nodes. We recommend increasing the size of the instance type when this metric reaches 60%.
MasterJVMMemoryPressure	Measures the percentage utilization of the JVM memory for the dedicated master nodes. We recommend increasing the size of the instance type when this metric reaches 85%.

For more information, see the following topics:

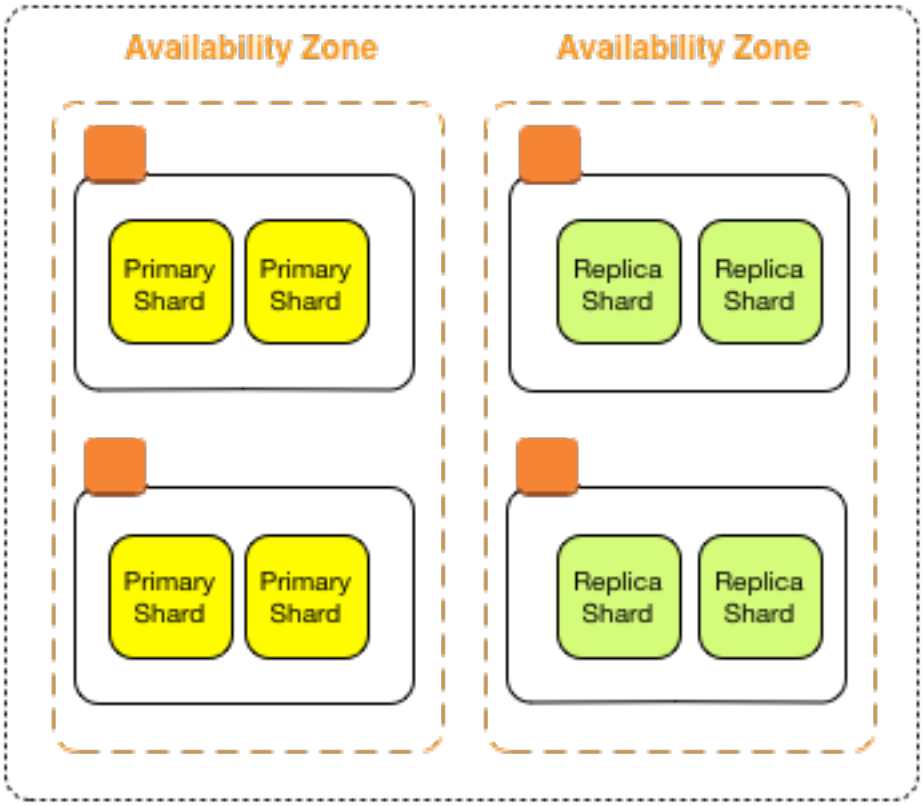
- Configuring an Amazon Elasticsearch Service Domain (p. 20) for instructions about how to add dedicated master nodes to an Elasticsearch cluster
- Monitoring Domains with Amazon CloudWatch (p. 42) for more information about monitoring CloudWatch metrics that are integrated into the Amazon Elasticsearch Service console
- Dedicated Master Nodes Resiliency in the Elasticsearch documentation for more information about dedicated master nodes

Enabling Zone Awareness (Console)

Each *region* is a separate geographic area with multiple, isolated locations known as *Availability Zones*. Zone awareness allocates the nodes and replica index shards belonging to an Elasticsearch cluster across two Availability Zones in the same region to prevent data loss and minimize downtime in the event of node and data center failure. If you enable zone awareness, you also must use the native Elasticsearch API to create replica shards for your cluster. Amazon ES will distribute the replicas across the nodes in the Availability Zones, which will increase the availability of your cluster. Enabling zone awareness for a cluster slightly increases network latencies.

> **Important**
> Zone awareness requires an even number of instances in the instance count. The default configuration for any index is a replica count of 1. If you specify a replica count of 0 for an index, zone awareness will not replicate the shards to the second Availability Zone. Without replica shards, there are no replicas to distribute to a second Availability Zone, and enabling the feature will not provide protection from data loss.

The following illustration shows a four-node cluster with zone awareness enabled. The service places all the primary index shards in one Availability Zone and all the replica shards in the second Availability Zone.



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.

Working with Manual Index Snapshots (AWS CLI)

Amazon Elasticsearch Service (Amazon ES) takes daily automated snapshots of the primary index shards in an Amazon ES domain, as described in Creating Automatic Index Snapshots (p. 31). However, you must contact the AWS Support team to restore an Amazon ES domain with an automated snapshot. If you need greater flexibility, you can take snapshots manually and manage them in a snapshot repository, an Amazon S3 bucket.

Manual snapshots provide a convenient way to migrate data across Amazon ES domains and to recover from failure. For more information, see Restoring to a different cluster in the Elasticsearch documentation. The service supports restoring indices and creating new indices from manual snapshots taken on both Amazon ES domains and self-managed Elasticsearch clusters.

Manual snapshots also allow you to address a red cluster service error (p. 69) yourself without waiting for AWS Support to restore the latest automatic snapshot. Your snapshot repository can hold several snapshots, each identified by a unique name. For a complete description of manual index snapshots, see Snapshot and Restore in the Elasticsearch documentation.

> **Note**
> If your Amazon ES domain experiences a red cluster (p. 69) error, AWS Support might contact you to ask whether you want to address the problem with your own manual index snapshots or you want the support team to restore the latest automatic snapshot of the domain. AWS Support will restore the latest automatic snapshot if you do not respond within seven days.

- Snapshot Prerequisites (p. 38)
- Registering a Snapshot Directory (p. 40)
- Taking Manual Snapshots (p. 41)
- Restoring Manual Snapshots (p. 41)

Snapshot Prerequisites

To create and restore 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.

Prerequisite	Description
S3 bucket	Stores manual snapshots for your Amazon ES domain. For more information, see Create a Bucket in the *Amazon S3 Getting Started Guide*.
IAM role	Delegates permissions to Amazon Elasticsearch Service. The trust relationship for the role must specify Amazon Elasticsearch Service in the `Principal` statement. The role type must be **Amazon EC2**. For instructions, see Create a Role to Delegate Permissions to an AWS Service in the IAM documentation. The IAM role is also required to register your snapshot repository with Amazon ES. Only IAM users with access to this role may register the snapshot repository. For more information, see Registering a Snapshot Repository (p. 40).
IAM policy	Specifies the actions that Amazon S3 may perform with your S3 bucket. The policy must be attached to the IAM role that delegates permissions to Amazon Elasticsearch Service. The policy must specify an S3 bucket in a `Resource` statement. For more information, see Creating Customer Managed Policies and Attaching Managed Policies in the *Using IAM Guide*.

S3 Bucket

Make a note of the Amazon Resource Name (ARN) for the S3 bucket where you will store manual snapshots. You will need it for the following:

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

The following example shows an ARN for an S3 bucket:

```
arn:aws:s3:::es-index-backups
```

For more information, see Create a Bucket in the *Amazon S3 Getting Started Guide*.

IAM Role

The role must specify Amazon Elasticsearch Service, `es.amazonaws.com`, in a `Service` statement in its trust relationship, as shown in the following example:

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

Note
Only IAM users with access to this role may register the snapshot repository.

For instructions, see Create a Role to Delegate Permissions to an AWS Service in the *Using IAM Guide*.

IAM Policy

An IAM policy must be attached to the role. The policy must specify the S3 bucket used to store manual snapshots for your Amazon ES domain. The following example specifies the ARN of the `es-index-backups` bucket:

```
 {
    "Version":"2012-10-17",
    "Statement":[
        {
            "Action":[
                "s3:ListBucket"
            ],
            "Effect":"Allow",
            "Resource":[
                "arn:aws:s3:::es-index-backups"
            ]
        },
        {
            "Action":[
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject",
                "iam:PassRole"
            ],
            "Effect":"Allow",
            "Resource":[
                "arn:aws:s3:::es-index-backups/*"
            ]
        }
    ]
}
```

For instructions, see Creating Customer Managed Policies and Attaching Managed Policies in the *Using IAM Guide*.

Registering a Snapshot Directory

As an IAM user with access to the new role, you must register the snapshot directory with Amazon Elasticsearch Service before you take manual index snapshots. This one-time operation requires that you sign your AWS request with the IAM role that grants permissions to Amazon ES.

> **Note**
> You cannot use `curl` to perform this operation because it does not support AWS request signing. Instead, use the sample Python client to register your snapshot directory.

Sample Python Client

Save the following sample Python code as a Python file, such as `snapshot.py`. Registering the snapshot directory with the service is a one-time operation. You can use `curl` to take subsequent snapshots, as described in Taking a Snapshot Manually (p. 41).

You must update the following in the sample code:

`region`
: AWS Region where you created the snapshot repository

`endpoint`
: Endpoint for your Amazon ES domain

`aws_access_key_id`
: IAM credential

`aws_secret_access_key`
: IAM credential

`path`
: Location of the snapshot repository

> **Note**
> The Python client requires that you install version 2.x of the `boto` package on the computer where you will register your snapshot repository.

```
from boto.connection import AWSAuthConnection

class ESConnection(AWSAuthConnection):

    def __init__(self, region, **kwargs):
        super(ESConnection, self).__init__(**kwargs)
        self._set_auth_region_name(region)
        self._set_auth_service_name("es")

    def _required_auth_capability(self):
        return ['hmac-v4']

if __name__ == "__main__":

    client = ESConnection(
            region='us-east-1',
            host='search-weblogs-etrt4mbbu254nsfupy6oiytuz4.us-
east-1.es.a9.com',
            aws_access_key_id='my-access-key-id',
            aws_secret_access_key='my-access-key', is_secure=False)
```

```
    print 'Registering Snapshot Repository'
    resp = client.make_request(method='POST',
            path='/_snapshot/weblogs-index-backups',
            data='{"type": "s3","settings": { "bucket": "es-index-
backups","region": "us-east-1","role_arn": "arn:aws:iam::123456789012:role/
MyElasticsearchRole"}}')
    body = resp.read()
    print body
```

Taking Manual Snapshots (AWS CLI)

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

- Name of your snapshot repository
- Name for the snapshot

To manually take a snapshot (AWS CLI)

- Run the following command to manually take a snapshot:

```
curl -XPUT 'http://<Elasticsearch_domain_endpoint>/
_snapshot/snapshot_repository/snapshot_name'
```

The following example takes a snapshot named `snapshot_1` and stores it in the `weblogs-index-backups` snapshot repository:

```
curl -XPUT 'http://<Elasticsearch_domain_endpoint>/_snapshot/weblogs-index-
backups/snapshot_1'
```

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: `BotoServerError: 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 'http://<Elasticsearch_domain_endpoint>/_snapshot/
<snapshot_repository>/_all?pretty'
```

Restoring Manual Snapshots (AWS CLI)

To restore a snapshot, perform the following procedure.

To manually restore a snapshot (AWS CLI)

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

   You cannot 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, then restore the snapshot
   - Restore the snapshot to a different Amazon ES domain

The following example demonstrates how to delete the existing indices for the *weblogs* domain:

```
curl -XDELETE 'http://search-weblogs-abcdefghijklmnojiu.us-east-1.a9.com/
_all'
```

2. Run the following command to restore a snapshot:

```
curl -XPOST 'http://<Elasticsearch_domain_endpoint>/
_snapshot/snapshot_repository/snapshot_name/_restore'
```

The following example restores `snapshot_1` from the `weblogs-index-backups` snapshot repository:

```
curl -XPOST 'http://search-weblogs-abcdefghijklmnojiu.us-east-1.a9.com/
_snapshot/weblogs-index-backups/snapshot_1/_restore'
```

Monitoring Cluster Metrics and Statistics with Amazon CloudWatch (Console)

An Elasticsearch cluster is a collection of one or more data nodes, optional dedicated master nodes, and storage required to run Elasticsearch and operate your Amazon ES domain. Each node in an Elasticsearch cluster automatically sends performance metrics to Amazon CloudWatch in one-minute intervals. Use the **Monitoring** tab in the Amazon Elasticsearch Service console to view these metrics, provided at no charge.

Statistics provide you with broader insight into each metric. For example, view the **Average** statistic for the **CPUUtilization** metric to compute the average CPU utilization for all nodes in the cluster. Each of the metrics falls into one of three categories:

- Cluster metrics (p. 43)
- Dedicated master node metrics (p. 46)
- EBS volume metrics (p. 47)

Tip
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. 43). 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 service provides the following metrics for clusters.

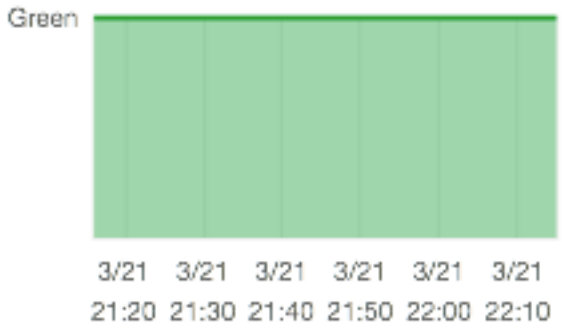
Tip
Use Amazon CloudWatch to check your cluster metrics if metrics are unavailable in the Amazon Elasticsearch Service console.

Cluster Metric	Description
ClusterStatus.green	Indicates that all index shards are allocated to nodes in a 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`. For more information, see Configuring Amazon Elasticsearch Service Domains (p. 20) in this guide and Update Indices Settings in the Elasticsearch documentation. Relevant statistics: Minimum, Maximum
ClusterStatus.red	Indicates that primary and replica shards of at least one index are not allocated to nodes in a cluster. A common cause for this state is a lack of free storage space on one or more of the data nodes in the cluster. In turn, a lack of free storage space prevents the service from distributing replica shards to the affected data node or nodes, and all new indices to start with a red cluster status. To recover, you must add EBS-based storage to existing data nodes, use larger instance types, or delete the indices and restore them from a snapshot. For recovery instructions, see Red Cluster Status (p. 69). Relevant statistics: Minimum, Maximum
Nodes	Specifies the number of nodes in a cluster. Relevant Statistics: Minimum, Maximum, Average
SearchableDocuments	Specifies the total number of searchable documents across all indices in a cluster. Relevant statistics: Minimum, Maximum, Average
DeletedDocuments	Specifies the total number of deleted documents across all indices in a cluster. Relevant statistics: Minimum, Maximum, Average
CPUUtilization	Specifies the maximum percentage of CPU resources used for data nodes in a cluster. Relevant statistics: Maximum, Average
FreeStorageSpace (Sum)	Specifies the free space, in megabytes, for all data nodes in a cluster. Relevant statistics: Sum

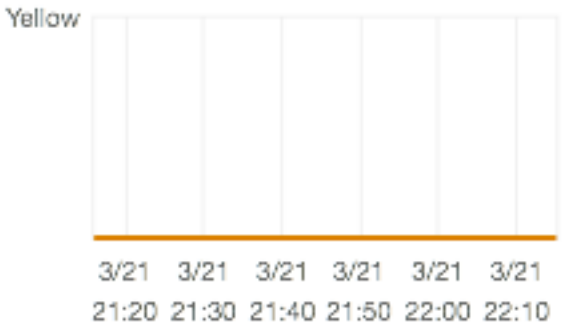
Cluster Metric	Description
FreeStorageSpace (Minimum)	Specifies the free space, in megabytes, for the single data node with the least available free space in a cluster. Amazon ES throws a `ClusterBlockException` when this metric reaches `0`. To recover, you must either delete indices, add larger instances, or add EBS-based storage to existing instances. For more information, see Recovering from a Lack of Free Storage Space (p. 70). Relevant statistics: Minimum
ClusterUsedSpace	Specifies the total used space, in megabytes, for a cluster. You can view this metric in the Amazon CloudWatch console, but not in the Amazon Elasticsearch Service console. Relevant statistics: Minimum, Maximum
JVMMemoryPressure	Specifies the maximum percentage of the Java heap used for all data nodes in the cluster. Relevant statistics: Maximum
AutomatedSnapshotFailure	Specifies 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	Specifies 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

The following screenshot shows the cluster metrics that are described in the preceding table.

ClusterStatus.green



ClusterStatus.yellow

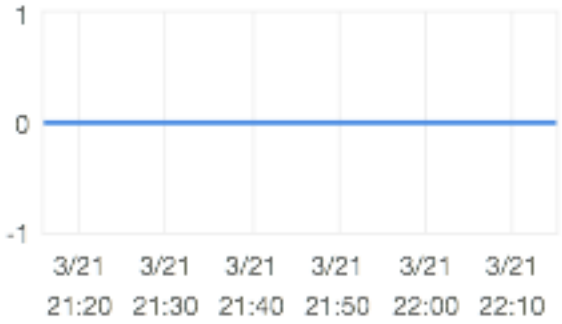


Nodes (Count)

Statistic: Average

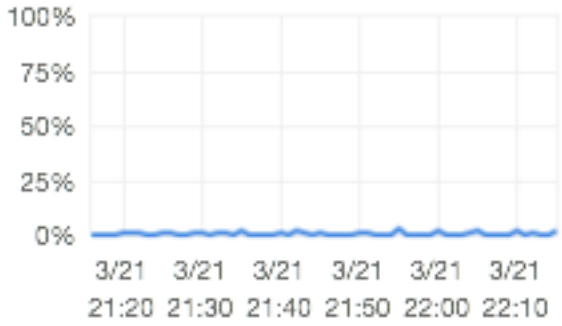
SearchableDocuments (Count)

Statistic: Average



CPUUtilization (Percent)

Statistic: Maximum

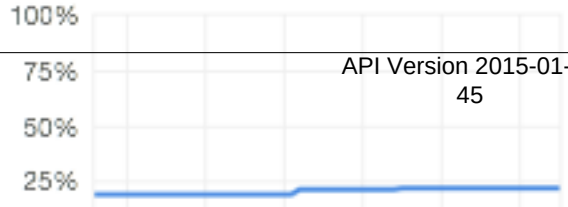


FreeStorageSpace (Megabytes)

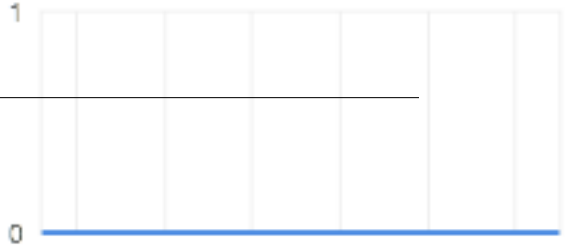
Statistic: Sum

JVMMemoryPressure (Percent)

Statistic: Maximum



AutomatedSnapshotFailure

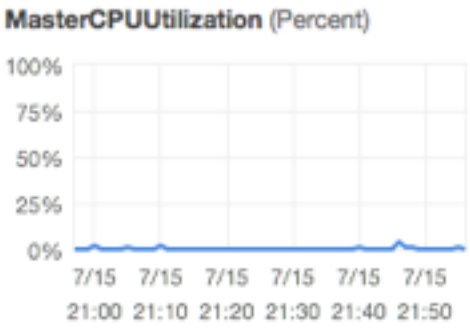


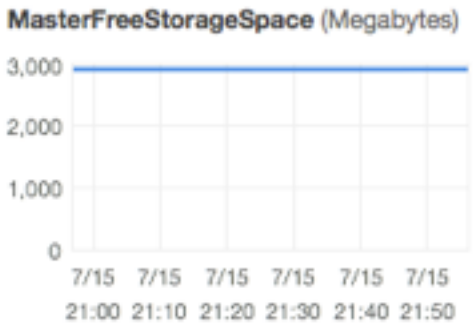
Dedicated Master Node Metrics

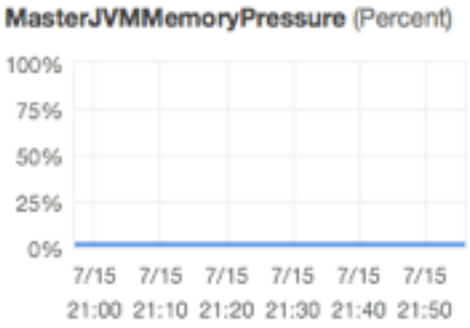
The service provides the following metrics for dedicated master nodes.

Dedicated Master Node Metric	Description
MasterCPUUtilization	Specifies 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%. Relevant statistics: Average
MasterFreeStorageSpace	This metric is not relevant and can be ignored. The service does not use master nodes as data nodes.
MasterJVMMemoryPressure	Specifies 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%. Relevant statistics: Maximum
MasterCPUCreditBalance	Specifies 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 Developer Guide*. This metric is available only for the t2.micro.elasticsearch, t2.small.elasticsearch, and t2.medium.elasticsearch instance types. Relevant Statistics: Minimum

The following screenshot shows the dedicated master nodes metrics that are described in the preceding table.







EBS Volume Metrics

The service provides the following metrics for EBS volumes.

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

EBS Volume Metric	Description
ReadIOPS	Specifies the number of input and output (I/O) operations per second for read operations on EBS volumes. Relevant statistics: Minimum, Maximum, Average
WriteIOPS	Specifies the number of input and output (I/O) operations per second for write operations on EBS volumes. Relevant statistics: Minimum, Maximum, Average

The following screenshot shows the EBS volume metrics that are described in the preceding table.

Auditing Amazon Elasticsearch Service Domains with AWS CloudTrail

Amazon Elasticsearch Service (Amazon ES) is integrated with AWS CloudTrail, a service that logs all AWS API calls made by, or on behalf of, your AWS account. The log files are delivered to an Amazon S3 bucket that you create and configure with a bucket policy that grants CloudTrail permissions to write log files to the bucket. CloudTrail captures all Amazon ES configuration service API calls, including those submitted by the Amazon Elasticsearch Service console.

You can use the information collected by CloudTrail to monitor activity for your search domains. You can determine the request that was made to Amazon ES, the source IP address from which the request was made, who made the request, and when it was made. To learn more about CloudTrail, including how to configure and enable it, see the *AWS CloudTrail User Guide*. To learn more about how to create and configure an S3 bucket for CloudTrail, see Amazon S3 Bucket Policy for CloudTrail.

Note
CloudTrail logs events only for configuration-related API calls to Amazon Elasticsearch Service. Data-related APIs are not logged.

The following example shows a sample CloudTrail log for Amazon ES:


```
{
    "Records": [
        {
            "eventVersion": "1.03",
            "userIdentity": {
                "type": "Root",
                "principalId": "000000000000",
                "arn": "arn:aws:iam::000000000000:root",
                "accountId": "000000000000",
                "accessKeyId": "A*****************A"
            },
            "eventTime": "2015-07-31T21:28:06Z",
            "eventSource": "es.amazonaws.com",
            "eventName": "CreateElasticsearchDomain",
            "awsRegion": "us-east-1",
            "sourceIPAddress": "Your IP",
            "userAgent": "es/test",
            "requestParameters": {
                "elasticsearchClusterConfig": {},
                "snapshotOptions": {
                    "automatedSnapshotStartHour": "0"
                },
                "domainName": "your-domain-name",
                "eBSOptions": {
                    "eBSEnabled": false
                }
            },
            "responseElements": {
                "domainStatus": {
                    "created": true,
                    "processing": true,
                    "aRN": "arn:aws:es:us-east-1:000000000000:domain/your-
domain-name",
                    "domainId": "000000000000/your-domain-name",
                    "elasticsearchClusterConfig": {
                        "zoneAwarenessEnabled": false,
```

```
                        "instanceType": "m3.medium.elasticsearch",
                        "dedicatedMasterEnabled": false,
                        "instanceCount": 1
                    },
                    "deleted": false,
                    "domainName": "your-domain-name",
                    "domainVersion": "1.5",
                    "accessPolicies": "",
                    "advancedOptions": {
                        "rest.action.multi.allow_explicit_index": "true"
                    },
                    "snapshotOptions": {
                        "automatedSnapshotStartHour": "0"
                    },
                    "eBSOptions": {
                        "eBSEnabled": false
                    }
                }
            },
            "requestID": "05dbfc84-37cb-11e5-a2cd-fbc77a4aae72",
            "eventID": "c21da94e-f5ed-41a4-8703-9a5f49e2ec85",
            "eventType": "AwsApiCall",
            "recipientAccountId": "000000000000"
        }
    ]
}
```

Amazon Elasticsearch Service Information in CloudTrail

When CloudTrail logging is enabled in your AWS account, API calls made to Amazon Elasticsearch Service (Amazon ES) operations are tracked in log files. Amazon ES records are written together with other AWS service records in a log file. CloudTrail determines when to create and write to a new file based on a time period and file size.

All Amazon ES configuration service operations are logged. For example, calls to `CreateElasticsearchDomain`, `DescribeElasticsearchDomain`, and `UpdateElasticsearchDomainConfig` generate entries in the CloudTrail log files. Every log entry contains information about who generated the request. The user identity information in the log helps you determine whether the request was made with root or IAM user credentials, with temporary security credentials for a role or federated user, or by another AWS service. For more information, see the `userIdentity` field in the CloudTrail Event Reference.

You can store your log files in your bucket indefinitely, or you can define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, your log files are encrypted using Amazon S3 server-side encryption (SSE). You can choose to have CloudTrail publish Amazon SNS notifications when new log files are delivered if you want to take quick action upon log file delivery. For more information, see Configuring Amazon SNS Notifications. You can also aggregate Amazon ES log files from multiple AWS Regions and multiple AWS accounts into a single Amazon S3 bucket. For more information, see Aggregating CloudTrail Log Files to a Single Amazon S3 Bucket.

Understanding Amazon Elasticsearch Service Log File Entries

CloudTrail log files contain one or more log entries where each entry is made up of multiple JSON-formatted events. A log entry represents a single request from any source and includes information

about the requested action, any parameters, the date and time of the action, and so on. The log entries are not guaranteed to be in any particular order—they are not an ordered stack trace of the public API calls. CloudTrail log files include events for all AWS API calls for your AWS account, not just calls to the Amazon ES configuration service API. However, you can read the log files and scan for `eventSource` `es.amazonaws.com`. The `eventName` element contains the name of the configuration service action that was called.

Visualizing Data with Kibana

Kibana is a popular open-source visualization tool designed to work with Elasticsearch. Amazon Elasticsearch Service (Amazon ES) provides a default installation of Kibana with every Amazon ES domain; you can find the Kibana endpoint on your domain dashboard in the service console. For more information about using Kibana to visualize your data, see the Kibana User Guide.

Connecting a Local Kibana Server to Amazon ES

Many customers have invested significant time configuring their own local Kibana servers. Rather than repeat that work with the default Kibana instance provided by Amazon ES, you can configure your local Kibana server to connect to the service by editing your `kibana.yml` configuration file with the following changes:

```
kibana_index: ".kibana-4"
elasticsearch_url: http:<elasticsearch_domain_endpoint>
```

You must add the `http:` prefix to your Amazon ES domain endpoint.

> **Tip**
> We recommend that you configure an IP-based access policy for the computer that hosts your local Kibana server. IP-based access policies restrict access to domain resources to one or more specific IP addresses. IP-based policies can also be configured to allow anonymous access, which enables you to submit unsigned requests to an Amazon ES domain. For more information, see Configuring Access Policies (p. 26).

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 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. You can find this service endpoint in the Amazon Elasticsearch Service console dashboard.

Configuration Field	Input | Output Plugin	Description
`region`	Input	Specifies the AWS Region where the Amazon S3 bucket resides.
`host`	Output	Specifies the service endpoint for the target Amazon ES domain.
`protocol`	Output	Specifies the protocol used to communicate with Amazon ES. Currently, the service supports HTTP only.
`ssl`	Output	Specifies whether to use SSL to connect to Amazon ES.
`port`	Output	Specifies the port used to connect with Amazon ES. Use port 443 when `ssl` is `true` and port 80 when `ssl` is `false`.
`flush_size`	Output	By default, Logstash fills a buffer with 5,000 events before sending the entire batch onward. However, if your documents are large, approaching 100 MB in size, we recommend configuring `flush_size` option to a larger value to prevent the buffer from filling too quickly. If you increase `flush_size`, we recommend also setting the Logstash `LS_HEAP_SIZE` environment variable to 2,048 MB to prevent running out of memory. For more information, see `flush_size` in the Elasticsearch documentation.

The following 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 on port 443

```
input{
    s3 {
        bucket => "wikipedia-stats-log"
        access_key_id => "lizards"
        secret_access_key => "lollipops"
        region => "us-east-1"
    }
}
output{
    elasticsearch {
        host => "search-logs-demo0-cpxczkdpi4bkb4c44g3csyln5a.us-
east-1.es.a9.com"
        protocol => "http"
        ssl => true
        port => 443
        flush_size => 250000
```

```
    }
}
```

The following example demonstrates the same configuration, but connects to Amazon ES without SSL on port 80:

```
input{
    s3 {
        bucket => "wikipedia-stats-log"
        access_key_id => "lizards"
        secret_access_key => "lollipops"
        region => "us-east-1"
    }
}
output{
    elasticsearch {
        host => "search-logs-demo0-cpxczkdpi4bkb4c44g3csyln5a.us-
east-1.es.a9.com"
        protocol => "http"
        ssl => false
        port => 80
        flush_size => 250000
    }
}
```

Note
The service request in the preceding example must be signed. For more information about signing requests, see Signing Amazon Elasticsearch Service Requests (p. 53). Use the logstash-output-amazon-es output plugin to sign and export Logstash events to Amazon ES. For instructions, see `README.md`.

Signing Amazon Elasticsearch Service Requests

If you're using a language for which AWS provides an SDK, we recommend that you use the SDK to submit Amazon Elasticsearch Service (Amazon ES) requests. All the AWS SDKs greatly simplify the process of signing requests, and save you a significant amount of time when compared with using the Amazon ES APIs directly. The SDKs integrate easily with your development environment and provide easy access to related commands.

If you choose to call the Amazon ES configuration service operations directly, you must sign your own requests. Configuration service requests must always be signed. Upload and search requests must be signed unless you configure anonymous access for those services. To sign a request, you calculate a digital signature using a cryptographic hash function, which returns a hash value based on the input. The input includes the text of your request and your secret access key. The hash function returns a hash value that you include in the request as your signature. The signature is part of the Authorization header of your request. After receiving your request, Amazon ES recalculates the signature using the same hash function and input that you used to sign the request. If the resulting signature matches the signature in the request, Amazon ES processes the request. Otherwise, the request is rejected.

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

Note
Amazon ES provides a Logstash output plugin to sign and export Logstash events to the service. Download the logstash-output-amazon-es plugin, and see `README.md` for instructions.

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 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 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 ten 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 a 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 can also 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 Cost Allocation and Tagging 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 the 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. 77), including the `AddTags`, `ListTags`, and `RemoveTags` commands. For more information about installing and using the AWS SDKs, see AWS Software Development Kits.

Loading Streaming Data into Amazon Elasticsearch Service

Amazon Elasticsearch Service (Amazon ES) supports integration with several AWS services, including streaming data from Amazon S3 buckets, Amazon Kinesis streams, Amazon DynamoDB Streams, and Amazon CloudWatch metrics. Most integrations use a Lambda function as an event handler in the cloud that responds to new data in Amazon S3 and Amazon Kinesis by processing it and streaming the data to your Amazon ES domain.

- Loading Streaming Data into Amazon ES from Amazon S3 (p. 58)
- Loading Streaming Data into Amazon ES from Amazon Kinesis (p. 61)
- Loading Streaming Data into Amazon ES from Amazon DynamoDB (p. 64)
- Loading Streaming Data into Amazon ES from Amazon CloudWatch (p. 64)

Streaming data provides fresh data for search and analytic queries. Amazon S3 pushes event notifications to AWS Lambda, as described in Using AWS Lambda with Amazon S3 in the *AWS Lambda Developer Guide*. Amazon Kinesis, however, requires AWS Lambda to poll for, or pull, event notifications, as described in Using AWS Lambda with Amazon Kinesis. You should be familiar with these service integrations before attempting to use them to load streaming data into your Amazon ES domain. For more information about these services, see the following AWS documentation:

- AWS Lambda Developer Guide
- Amazon S3 Developer Guide
- Amazon Kinesis Developer Guide
- Amazon DynamoDB Developer Guide
- Amazon CloudWatch Developer Guide

Note
AWS Lambda is available in limited regions. For more information, see the list of AWS Lambda regions in the *AWS General Reference*.

Loading Streaming Data into Amazon ES from Amazon S3

You can integrate your Amazon ES domain with Amazon S3 and AWS Lambda so that any new data sent to an S3 bucket triggers an event notification to Lambda, which then runs your custom Java or Node.js application code. After your application processes the data, it streams the data to your Amazon ES domain. At a high level, setting up to load streaming data to Amazon ES requires the following steps:

1. Creating a Lambda deployment package (p. 59)
2. Configuring a Lambda function (p. 60)
3. Granting authorization to add data to your Amazon ES domain (p. 60)

You must also create an Amazon S3 bucket and an Amazon ES domain. Setting up this integration path requires the following prerequisites.

Prerequisite	Description
Amazon S3 Bucket	The event source that triggers your Lambda function. For more information, see Create a Bucket in the *Amazon Simple Storage Service Getting Started Guide*. The bucket must reside in the same AWS Region as your Amazon ES domain.
Amazon ES Domain	The destination for data after it is processed by the application code in your Lambda function. For more information, see Creating Amazon Elasticsearch Service Domains (p. 15) in this guide.
Lambda Function	The Java or Node.js application code that runs when S3 pushes an event notification to Lambda. Amazon ES provides a sample application in Node.js, `s3_lambda_es.js`, that you can download to get started: Lambda Sample Code for Amazon ES.
Lambda Deployment Package	A .zip file consisting of your Java or Node.js application code and any dependencies. For information about the required folder hierarchy, see Creating a Lambda Deployment Package (p. 59) in this guide. For general information about creating Lambda deployment packages, see Creating a Deployment Package (Node.js) and Creating a Deployment Package (Java).
Amazon ES Authorization	Your integration must grant authorization by means of an IAM access policy that permits Lambda to add data to your Amazon ES domain. Attach this policy to the Amazon S3 execution role that you create as part of your Lambda function. For details, see Granting Authorization to Add Data to Your Amazon ES Domain (p. 60).

Setting Up to Load Streaming Data into Amazon ES from Amazon S3

This section provides additional details about setting up the prerequisites for loading streaming data into Amazon ES from Amazon S3. After you finish configuring the integration, data streams automatically to your Amazon ES domain whenever new data is added to your Amazon S3 bucket.

Creating a Lambda Deployment Package

Create a .zip file that contains your Lambda application code and any dependencies.

To create a deployment package:

1. Create a directory structure like the following:

   ```
   eslambda
       \node_modules
   ```

   You can use any name for the top-level folder, rather than `eslambda`. However, the subfolder must be named `node_modules`.
2. Place your application source code in the `eslambda` folder.
3. Add or edit the following four global variables:
   - `endpoint`, the Amazon ES domain endpoint.
   - `region`, the AWS Region in which you created your Amazon ES domain.
   - `index`, the name of the Amazon ES index to use for data streamed from Amazon S3.
   - `doctype`, the Amazon ES document type of the streamed data. For more information, see Mapping Types in the Elasticsearch documentation.

   The following example from `s3_lambda_es.js` configures the sample application to use the `streaming-logs` Amazon ES domain endpoint in the us-east-1 AWS Region:

   ```
   /* Globals */
   var esDomain = {
       endpoint: 'search-streaming-logs-okga24ftzsbz2a2hzhsqw73jpy.us-
   east-1.es.a9.com',
       region: 'us-east-1',
       index: 'streaming-logs',
       doctype: 'apache'
   };
   ```
4. Install any dependencies required by your application.

   For example, if you are using Node.js, you must execute the following command for each `require` statement in your application code:

   ```
   npm install <dependency>
   ```
5. Verify that all runtime dependencies that are required by your application code are located in the `node_modules` folder.
6. Execute the following command to package the application code and dependencies:

   ```
   zip -r eslambda.zip *
   ```

   The name of the zip file must match the top-level folder.

For more information about creating Lambda deployment packages, see Create a Lambda Function Deployment Package (Node.js) and Creating a Deployment Package (Java).

Configuring a Lambda Function

Use AWS Lambda to create and configure your Lambda function. Although the instructions in the Using AWS Lambda with Amazon S3 tutorial use the AWS CLI, you can perform all configuration settings described in the following table with the AWS Lambda console.

> **Note**
> For more information about creating and configuring a Lambda function, see the AWS Lambda Developer Guide.

Function Configuration	Description
IAM Execution Role	The name of the IAM role used to execute actions on Amazon S3. While creating your Lambda function, the Lambda console automatically opens the IAM console to help you create the execution role. Later, you must also attach an IAM access policy to this role that permits Lambda to add data to your Amazon ES domain. For details, see Granting Authorization to Add Data to Your Amazon ES Domain (p. 60).
Event Source	Specify the S3 bucket as the event source for the Lambda function. For instructions, see the Using AWS Lambda with Amazon S3 tutorial. AWS Lambda will automatically add the necessary permissions for Amazon S3 to invoke your Lambda function from this event source. Optionally, specify a file suffix to filter what kinds of files trigger the Lambda function, such as `.log`.
Handler	The name of the file containing the application source code, but with the `.handler` file suffix. For example, if your application source code resides in a file named `s3_lambda_es.js`, the handler must be configured as `s3_lambda_es.handler`. For more information, see Getting Started in the *AWS Lambda Developer Guide*. Amazon ES provides a sample application in Node.js that you can download to get started: Lambda Sample Code for Amazon ES.
Timeout	The length of time Lambda should wait before canceling an invocation request. The default value of three seconds is too short for the Amazon ES use case. We recommend configuring your timeout for 10 seconds.

For more function configuration details, see Configuring a Lambda Function (p. 60) in this guide. For general information, see the following topics in the *AWS Lambda Developer Guide*:

- Authoring Lambda Functions in Node.js
- Authoring Lambda Functions in Java

Granting Authorization to Add Data to Your Amazon ES Domain

When you choose **S3 Execution Role** as the IAM role to execute actions on S3, Lambda opens the IAM console and helps you to create a new execution role. Lambda automatically adds the necessary permissions to invoke your Lambda function from this event source. After you create the role, open it in the IAM console and attach the following IAM access policy to the role so that Lambda has permission to stream data to Amazon ES:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:us-west-2:123456789012:domain/streaming-
logs/*"
        }
    ]
}
```

For more information about attaching IAM access policies to roles, see Tutorial: Create and Attach Your First Customer Managed Policy in the *IAM User Guide*.

Loading Streaming Data into Amazon Elasticsearch Service from Amazon Kinesis

Amazon Elasticsearch Service (Amazon ES) also supports loading streaming data from Amazon Kinesis. This integration also relies on AWS Lambda as an event handler in the cloud. Amazon Kinesis requires Lambda to poll your Amazon Kinesis stream to determine whether it has new data that will automatically invoke your Lambda function. After your Lambda function finishes processing any new data, it streams the data to your Amazon ES domain.

At a high level, setting up to stream data to Amazon ES requires the following steps:

1. Creating a Lambda deployment package (p. 62).
2. Configuring a Lambda function (p. 63).
3. Granting authorization to add data to your Amazon ES domain (p. 64).

You must also create an Amazon Kinesis stream and an Amazon ES domain. Setting up this integration path requires the following prerequisites.

Prerequisite	Description
Amazon Kinesis Stream	The event source for your Lambda function. For instructions about creating Amazon Kinesis streams, see Amazon Kinesis Streams.
Elasticsearch Domain	The destination for data after it is processed by the application code in your Lambda function. For more information, see Creating Amazon ES Domains (p. 15) in this guide.
Lambda Function	The Java or Node.js application code that runs when Amazon Kinesis pushes an event notification to Lambda. Amazon ES provides a sample application in Node.js, `kinesis_lambda_es.js`, that you can download to get started: Lambda Sample Code for Amazon ES.
Lambda Deployment Package	A .zip file that consists of your Java or Node.js application code and any dependencies. For more information about creating Lambda deployment packages, see Creating a Deployment Package (Node.js) and Creating a Deployment Package (Java).

Prerequisite	Description
Amazon ES Authorization	Your integration must grant IAM access policy that permits Lambda to add data to your Amazon ES domain. Attach this policy to the Amazon Kinesis execution role that you create as part of your Lambda function. For details, see Granting Authorization to Add Data to Your Amazon ES Domain (p. 60).

Setting Up to Load Streaming Data into Amazon ES from Amazon Kinesis

This section provides further details about setting up the prerequisites for loading streaming data into Amazon ES from Amazon Kinesis. After you finish configuring the integration, Lambda automatically streams data to your Amazon ES domain whenever new data is added to your Amazon Kinesis stream.

Creating a Lambda Deployment Package

Create a .zip file containing your Lambda application code and any dependencies.

To create a deployment package:

1. Create a directory structure like the following:

```
eslambda
    \node_modules
```

   You can use any name for the top-level folder, rather than `eslambda`. However, you must name the subfolder `node_modules`.
2. Place your application source code in the `eslambda` folder.
3. Add or edit the following global variables in your sample application:

   - `endpoint`, the Amazon ES domain endpoint.
   - `region`, the AWS Region in which you created your Amazon ES domain.
   - `index`, the name of the Amazon ES index to use for data streamed from Amazon Kinesis.
   - `doctype`, the Amazon ES document type of the streamed data. For more information, see Mapping Types in the Elasticsearch documentation.

   The following example from the `kinesis_lambda_es.js` configures the sample application to use the `streaming-logs` Amazon ES domain endpoint in the us-east-1 AWS Region.

```
/* Globals */
var esDomain = {
    endpoint: 'search-streaming-logs-okga24ftzsbz2a2hzhsqw73jpy.us-
east-1.es.a9.com',
    region: 'us-east-1',
    index: 'streaming-logs',
    doctype: 'apache'
};
```

4. Install any dependencies that are required by your application.

   For example, if you are using Node.js, you must execute the following command for each `require` statement in your application code:

```
npm install <dependency>
```

5. Verify that all runtime dependencies that are required by your application code are located in the `node_modules` folder.
6. Execute the following command to package the application code and dependencies:

```
zip -r eslambda.zip *
```

   The name of the zip file must match the top-level folder.

For more information about creating Lambda deployment packages, see Create a Lambda Function Deployment Package (Node.js) and Creating a Deployment Package (Java).

Configuring a Lambda Function

Use AWS Lambda to create and configure your Lambda function. Although the tutorial instructions in Using AWS Lambda with Amazon Kinesis use the AWS CLI, you can perform all configuration settings described in the following table with the **Configuration** tab in the AWS Lambda console.

> **Note**
> For more information about creating and configuring a Lambda function, see Getting Started: Authoring AWS Lambda Code in Java and Getting Started: Authoring AWS Lambda Code in Node.js.

Configuration	Description
Amazon Kinesis stream	The event source of your Lambda Function. For instructions, see Amazon Kinesis Streams.
IAM execution role	The name of the IAM role used to execute actions on Amazon Kinesis. While configuring your Lambda function, the Lambda console automatically opens the IAM console to help you create the execution role. Later, you must also attach an IAM access policy to this role that permits Lambda to send data to your Amazon ES domain. You must also attach an IAM access policy to this role that permits Lambda to send data to your Amazon ES domain. For details, see Granting Authorization to Add Data to Your Amazon ES Domain (p. 64).
Handler	The name of the file that contains the application source code, but with the `.handler` file suffix. For example, if your application source code resides in a file named `kinesis_lambda_es.js`, the handler must be configured as `kinesis_lambda_es.handler`. For more information, see Getting Started 1: Invoking Lambda Functions from User Applications Using the AWS Lambda Console (Node.js). Amazon ES provides a sample application in Node.js that you can download to get started: Lambda Sample Code for Amazon ES.
Timeout	The length of time Lambda should wait before canceling an invocation request. The default value of three seconds is too short for this use case. We recommend configuring your timeout for 10 seconds.

For more information, see the following topics in the *AWS Lambda Developer Guide*:

- Authoring Lambda Functions in Node.js
- Authoring Lambda Functions in Java

Granting Authorization to Add Data to Your Amazon ES Domain

When you choose **Kinesis Execution Role** as the IAM role to execute actions on Amazon Kinesis, Lambda opens the IAM console and requires you to create a new execution role. Lambda automatically adds the necessary permissions to invoke your Lambda function from this event source. After you create the new role, open it in the IAM console and attach the following IAM access policy to the new role so that Lambda has permission to stream data to Amazon ES:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:us-west-2:123456789012:domain/streaming-
logs/*"
        }
    ]
}
```

For more information about attaching IAM access policies to roles, see Tutorial: Create and Attach Your First Customer Managed Policy in the *IAM User Guide*.

Loading Streaming Data into Amazon ES from Amazon DynamoDB

Amazon ES supports streaming data to your Amazon ES domain from Amazon DynamoDB Streams using the DynamoDB Logstash input plugin and the Amazon ES Logstash output plugin. For instructions, see Logstash Plugin for Amazon DynamoDB in the *Amazon DynamoDB Developer Guide*.

Loading Streaming Data into Amazon ES from Amazon CloudWatch

Amazon ES supports streaming data to your Amazon ES domain from CloudWatch Logs through a CloudWatch Logs subscription. For more information about Amazon CloudWatch subscriptions, see Real-time Processing of Log Data with Subscriptions. For configuration information, see this topic in the *Amazon CloudWatch Developer Guide.*

Migrating Data in a Domain from Elasticsearch Version 1.5 to Version 2.3

Amazon Elasticsearch Service (Amazon ES) currently supports two Elasticsearch versions: 1.5 and 2.3. Compared to Elasticsearch version 1.5, version 2.3 offers improved performance, memory management, and security. It also offers several additional features, including the following:

- Pipeline aggregations to perform advanced analytics, such as moving averages and derivatives
- Enhancements to geospatial queries
- Configurable store compression

For more information about the differences between Elasticsearch 1.5 and 2.3, see the Elasticsearch documentation. For information about the Elasticsearch APIs that Amazon ES supports for both 1.5 and 2.3, see Supported Elasticsearch Operations (p. 67).

If you are starting a new Elasticsearch project, we strongly recommend that you choose version 2.3. If you have an existing 1.5 domain, you have the option to migrate your data to a new 2.3 domain. The migration is necessary because you can't change the version of an existing domain. You can also choose to simply keep your 1.5 domain without migrating your data to a 2.3 domain.

To migrate your data, perform the following procedure.

To migrate your data from a 1.5 domain to a 2.3 domain

1. Use the /_plugin/migration Elasticsearch plugin to determine whether you can directly upgrade to version 2.3 or need to make changes to your data before migration.
   a. In a web browser, open http://*`Elasticsearch_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. For more information, see Elasticsearch breaking changes.
2. Create a manual snapshot of your 1.5 domain. For more information, see Working with Manual Index Snapshots (p. 37).
3. Create a 2.3 domain. For more information, see Creating Amazon ES Domains (p. 15).

4. Copy the snapshot data from the 1.5 domain to the 2.3 domain. For more information, see Working with Manual Index Snapshots (p. 37).
5. Test your application thoroughly.
6. (Optional) Delete your 1.5 domain if you no longer need it. Otherwise, you will continue to incur charges for the domain.

Supported Elasticsearch Operations

Amazon Elasticsearch Service (Amazon ES) currently supports Elasticsearch versions 1.5 and 2.3. For Elasticsearch 1.5, Amazon ES supports the following operations:

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`
- `/_cluster/health`
- `/_cluster/settings` for three properties (PUT only):
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_cluster/stats`
- `/_count`
- `/_flush`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_nodes`
- `/_plugin/kibana`
- `/_plugin/kibana3`
- `/_percolate`
- `/_refresh`
- `/_search`
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_template`

For more information about Elasticsearch 1.5 operations, see the Elasticsearch documentation.

Note
The `/_cluster/settings` operation does not support the HTTP GET method. You can set three cluster settings, but cannot retrieve them. For more information about using `/_cluster/settings` to set JVM-based memory breakers for Amazon ES, see JVM OutOfMemory Error (p. 71) in this guide. For more information about the three supported cluster settings, see Cluster Update Settings and Field Data in the Elasticsearch documentation.

For Elasticsearch 2.3, Amazon ES supports the following operations:

- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`
- `/_cluster/stats`
- `/_count`
- `/_flush`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_percolate`
- `/_refresh`
- `/_search`
- `/_snapshot`
- `/_stats`
- `/_status`

• `/_cluster/health`
• `/_cluster_settings` for three properties (PUT only):
  • `indices.breaker.fielddata.limit`
  • `indices.breaker.request.limit`
  • `indices.breaker.total.limit`
	• `/_nodes`
	• `/_plugin/kibana`
		• `/_template`

For more information about Elasticsearch 2.3 operations, see the Elasticsearch documentation.

Note

The `/_cluster/settings` operation does not support the HTTP GET method. You can set three cluster settings, but cannot retrieve them. For more information about using `/_cluster/settings` to set JVM-based memory breakers for Amazon ES, see JVM OutOfMemory Error (p. 71) in this guide. For more information about the three supported cluster settings, see Cluster Update Settings and Field Data in the Elasticsearch documentation.

Handling AWS Service Errors

This chapter describes how to respond to common AWS service errors. Consult this chapter before contacting AWS Support.

Failed Cluster Nodes

EC2 instances might experience unexpected terminations and restarts. Typically, Amazon ES restarts the nodes for you. However, it is possible for one or more nodes in an Elasticsearch cluster to remain in a failed condition. You can detect this status by watching the **Nodes** CloudWatch metric on the **Monitoring** tab of your domain dashboard in the Amazon ES console to see if the reported number of nodes is fewer than the number that you configured for your cluster. If the metric indicates that one or more nodes is down for more than one day, contact AWS Support.

> **Note**
> The **Nodes** metric is not accurate during changes to your cluster configuration and during routine maintenance for the service. This is expected behavior. The metric will report the correct number of cluster nodes soon.

You can protect your clusters from unexpected node terminations and restarts by creating at least one replica for each index in your Amazon ES domain. For more information, see Shards and Replicas in the Elasticsearch documentation.

Red Cluster Status

A red cluster status indicates that both primary and replica shards are not allocated to nodes in your cluster, as described in Monitoring Domains with Amazon CloudWatch Metrics (p. 42). No additional automatic snapshots are taken, even of healthy indices, while the red cluster status persists. This state is commonly caused by the following:

- Data nodes in the Elasticsearch cluster lack free storage space on data nodes
- The Elasticsearch process crashed due to a continuous heavy processing load on a data node

You must take action if the red cluster status is due to a lack of free storage space or a heavy processing load.

Recovering from a Lack of Free Storage Space

You should actively monitor the **FreeStorageSpace** cluster metric to ensure that the data nodes in an Elasticsearch cluster do not run out of free storage space. Amazon ES throws a `ClusterBlockException` when this metric reaches `0`, and the affected cluster is unable to process configuration changes and will not benefit from software upgrades to the service.

To resolve red cluster status caused by red indices

1. Use the `/_cat/indices` Elasticsearch API to determine which of the indices in your Amazon ES domain are currently unassigned to nodes in your cluster:

   ```
   curl -XGET 'http://<Elasticsearch_domain_endpoint>/_cat/indices
   ```

   For more information, see cat indices in the Elasticsearch documentation.
2. Add EBS-based storage, use larger instance types, or delete data from the affected indices. For more information, see Configuring EBS Storage (p. 23) and Configuring Amazon ES Domains (p. 20).

If you prefer to receive a notification of a red cluster status rather than checking the **Monitoring** tab for your domain in the Amazon ES console, you can create an Amazon CloudWatch alarm to notify you when your cluster enters the red status. For more information, see Creating Amazon CloudWatch Alarms in the *Amazon CloudWatch Developer Guide*.

Recovering from a Continuous Heavy Processing Load

You should monitor the following cluster metrics to determine if a red cluster state is due to a continuous heavy processing load on a data node.

Relevant Metric	Description	Recovery
JVMMemoryPressure	Specifies the percentage of the Java heap used for all data nodes in the Elasticsearch cluster. View the **Maximum** statistic for this metric and look for oscillations that indicate a pattern of heavy memory pressure, likely due to complex queries or large data fields.	Set memory circuit breakers for the JVM. For more information, see JVM OutOfMemoryError (p. 71).
CPUUtilization	Specifies the percentage of CPU resources used for data nodes in the cluster. View the **Maximum** statistic for this metric and look for a continuous pattern of high usage.	Increase the number of data nodes or increase the size of the instance types of existing data nodes. For more information, see Configuring Amazon ES Domains (p. 20).
Nodes	Specifies the number of nodes in the Elasticsearch cluster. View the **Minimum** statistic for this metric. This value fluctuates when the service deploys a new fleet of instances for a cluster.	Increase the number of data nodes. For more information, see Configuring Amazon ES Domains (p. 20).

Yellow Cluster Status

A yellow cluster 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 data node to which the service can assign a replica. Increase your node count to obtain a green cluster status. For more information, see Configuring Amazon ES Domains (p. 20) in this guide and Update Indices Settings in the Elasticsearch documentation.

ClusterBlockException

You might receive a `ClusterBlockException` error for the following issues.

Logstash with Zero FreeStorageSpace

Logstash might throw a `ClusterBlockException` for many reasons, including a lack of storage space to which it can write. If you receive a `ClusterBlockException` error from Logstash while loading bulk data to your Elasticsearch cluster, check the **FreeStorageSpace** metric on the **Monitoring** tab of your domain dashboard in the Amazon ES console to see if your cluster has run out of storage space. If the available free storage space is `0`, either configure your cluster to use a larger instance type or add an EBS volume for additional storage. For more information, see Configuring Amazon ES Domains (p. 20) and Configuring EBS Storage (p. 23).

Block Disks due to Low Memory

For t2 instances, when the **JVMMemoryPressure** metric exceeds 92%, Amazon ES triggers a protection mechanism by blocking all write operations to prevent the cluster from getting into red status. When the protection is on, write operations will fail with a `ClusterBlockException` error, new indexes cannot be created, and the `IndexCreateBlockException` error will be thrown.

JVM OutOfMemoryError

A JVM `OutOfMemoryError` typically indicates 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` For more information, see Cluster Update Settings in the Elasticsearch documentation.
Field Data Breaker	Percentage of JVM heap memory allowed to load a single data field into memory. The default value is 60%. We recommend raising this limit if you are uploading data with large fields.	`indices.breaker.fielddata.limit` For more information, see Field data in the Elasticsearch documentation.

Circuit Breaker	Description	Cluster Setting Property
Request Breaker	Percentage of JVM heap memory allowed for data structures used to respond to a service request. The default value is 40%. We recommend raising this limit if your service requests involve calculating aggregations.	`indices.breaker.request.limit` For more information, see Field data in the Elasticsearch documentation.

Note
The `/_cluster/settings` operation does not support the HTTP GET method. You can set three cluster settings, but cannot retrieve them.

Troubleshooting

The following sections offer solutions to common problems that you might encounter when you use services (such as Amazon S3, Amazon Kinesis, and IAM) and products (such as Kibana) that integrate with Amazon Elasticsearch Service (Amazon ES):

- Kibana: I cannot sign AWS service requests to the Kibana service endpoint (p. 73)
- Kibana: I don't see the indices for my Elasticsearch domain in Kibana 4 (p. 74)
- Kibana: I get a browser error when I use Kibana to view my data (p. 75)
- Integrations: I don't see a service role for Amazon ES in the IAM console (p. 76)

For information about service-specific errors, see Handling AWS Service Errors (p. 69) in this guide.

Kibana: I Cannot Sign AWS Service Requests to the Kibana Service Endpoint

The Kibana endpoint does not support signed AWS service requests. We recommend that you access Kibana with one of the configuration options described in the following table.

Kibana Configuration	Description
Anonymous, IP-based Access	If the Kibana host is behind a firewall, configure the Kibana endpoint to accept anonymous requests from the IP address of the firewall. Use CIDR notation if you need to specify a range of IP addresses.
NAT Gateway with Amazon VPC	Amazon VPC supports NAT gateways. When you create a NAT gateway, you must specify an Elastic IP address to associate with the gateway. The NAT gateway sends traffic to the public Internet gateway using the Elastic IP address as the source IP address. Specify this Elastic IP address in the access policy for Kibana to allow all requests from the gateway. For more information, see NAT Gateway Basics and Creating a NAT Gateway in the *Amazon Virtual Private Cloud User Guide*.

Examples

The following example is an anonymous, IP-based access policy that specifies a range of IP addresses and an 18-bit routing prefix:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-2:123456789012:domain/mydomain/_plugin/
kibana",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "192.240.192.0/18"
        }
      }
    }
  ]
}
```

The following example specifies anonymous access from the Elastic IP address associated with a NAT gateway:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-2:123456789012:domain/mydomain/_plugin/
kibana",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "198.51.100.4"
        }
      }
    }
  ]
}
```

Kibana: I Don't See the Indices for My Elasticsearch Domain in Kibana 4

Users familiar with Kibana 3 but new to Kibana 4 sometimes have difficulty finding their indices in the interface. Unlike Kibana 3, which provides default dashboards to view your data, Kibana 4 requires

you to first specify an index name or pattern that matches the name of an index in your Amazon ES domain. For example, if your Amazon ES domain contains an index named `movies-2013`, any of the following patterns would match this index:

- movies-2013
- movies-*
- mov*

You can configure visualizations for the data in your Amazon ES domain index after you specify the index name or pattern. You can find the names of your domain indices on the **Indices** tab in the Amazon ES console. For more information about using Kibana 4, see the Kibana User Guide.

Tip
Amazon ES also supports Kibana 3, so you can configure access to that version of the tool if you prefer to use it. Specify `/_plugin/kibana3` as the resource in your access policy rather than `/_plugin/kibana`. After the service finishes processing the configuration change, you can access Kibana 3 by manually editing the Kibana service endpoint provided in the Amazon ES console. For example, if the console indicates that your Kibana endpoint is `mydomain-6w5y8xjt5ydwsrubmdk4m5kcpa.us-west-2.es.amazonaws.com/_plugin/kibana/`, point your browser to `mydomain-6w5y8xjt5ydwsrubmdk4m5kcpa.us-west-2.es.amazonaws.com/_plugin/kibana3/` instead.

Kibana: I Get a Browser Error When I Use Kibana to View My Data

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.

   For example, the following service error message indicates that a search request likely failed for one of the reasons shown in the following table:

   ```
   "Request to Elasticsearch failed: {"error":"SearchP…be larger than limit of [5143501209/4.7gb]]; }]"}"
   ```

Potential Cause	Workaround
You reached the JVM request memory circuit breaker.	The request breaker specifies the percentage of JVM memory used to respond to a service request. You can configure JVM circuit breakers to work around this failure. For more information about configuring JVM circuit breakers, see JVM OutOfMemoryError (p. 71) in Handling AWS Service Errors (p. 69) in this guide.

Potential Cause	Workaround
You specified a generic regular expression in your Kibana dashboard, such as `logstash*`.	Use a more restrictive regular expression, such as limiting results to a subset of indices over a time period of the last seven days.

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.

Integrations: I Don't See a Service Role for Amazon ES in the IAM Console

You can integrate Amazon ES with other services, such as Amazon S3 and Amazon Kinesis, as described in Loading Streaming Data into Amazon ES from Amazon S3 (p. 58) and Loading Streaming Data into Amazon ES from Amazon Kinesis (p. 61) in this guide. Both of these integrations use AWS Lambda as an event handler in the cloud. When you create a Lambda function using the AWS Lambda console, the console automatically opens the IAM console to help you create the required execution role. You do not need to open the IAM console yourself and select a service role. However, you must open the IAM console after AWS Lambda helps you to create the new role and attach the following IAM policy to it:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "es:*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:es:us-west-2:123456789012:domain/
<my_domain_name>/*"
        }
    ]
}
```

For step-by-step procedures, see Loading Streaming Data into Amazon ES from Amazon S3 (p. 58) and Loading Streaming Data into Amazon ES from Amazon Kinesis (p. 61) in this guide.

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. 15).

- Actions (p. 77)
- Data Types (p. 97)
- Errors (p. 105)

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. 53) in this guide and Signature Version 4 Signing Process in the *AWS General Reference*.

Action	HTTP Method
`AddTags` (p. 78)	`POST`
`CreateElasticsearchDomain` (p. 79)	`POST`
`DeleteElasticsearchDomain` (p. 82)	`DELETE`
`DescribeElasticsearchDomain` (p. 84)	`GET`
`DescribeElasticsearchDomains` (p. 85)	`POST`

Action	HTTP Method
DescribeElasticsearchDomainConfig (p. 89)	GET
ListDomainNames (p. 88)	GET
ListTags (p. 91)	GET
RemoveTags (p. 92)	POST
UpdateElasticsearchDomainConfig (p. 93)	POST

AddTags

Attaches resource tags to an Amazon ES domain. Use the POST HTTP method with this operation. For more information, see Tagging Amazon ES Domains (p. 54).

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[1]	Description
TagList	TagList (p. 105)	Yes	List of resource tags
ARN	ARN (p. 98)	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 /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. Use the HTTP `POST` method with this operation. For more information, see Creating Amazon ES Domains.

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>"
    },
    "AccessPolicies": "<ACCESS_POLICY_DOCUMENT>",
    "SnapshotOptions": {
        "AutomatedSnapshotStartHour": <START_HOUR>
    },
    "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "<TRUE|FALSE>",
            "indices.fielddata.cache.size": "<PERCENTAGE_OF_HEAP>"
    }
}
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Parameter	Data Type	Required	Description
`DomainName`	`DomainName` (p. 99)	Yes	Name of the Amazon ES domain to create.
`ElasticsearchVersion`	String	No	Version of Elasticsearch. Currently, Amazon ES supports 1.5 and 2.3. If not specified, 1.5 is used as the default.
`ElasticsearchClusterConfig`	`ElasticsearchClusterConfig` (p. 100)	No	Container for the cluster configuration of an Amazon ES domain.
`EBSOptions`	`EBSOptions` (p. 99)	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. 23).
`AccessPolicies`	String	No	IAM policy document specifying the access policies for the new Amazon ES domain. For more information, see Configuring Access Policies (p. 26).
`SnapshotOptions`	`SnapshotOptionsStatus` (p. 104)	No	Container for parameters required to configure automated snapshots of domain indices. For more information, see Configuring Snapshots (p. 31).
`AdvancedOptions`	`AdvancedOptionsStatus` (p. 97)	No	Key-value pairs to specify the following advanced options: • `rest.action.multi.allow_explicit_index` • `indices.fielddata.cache.size`

Parameter	Data Type	Required	Description
			For more information, see Configuring Advanced Options (p. 32).

Response Elements

Field	Data Type	Description
`DomainStatus`	ElasticsearchDomainStatus (p. 193)	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

The following example demonstrates the following:

- Creates an Amazon ES domain named `streaming-logs`
- Configures a cluster with six instances of the m3.medium.elasticsearch instance type and three dedicated master nodes of the same type
- Enables zone awareness

Request

```
POST /2015-01-01/es/domain
{
    "DomainName": "streaming-logs",
    "ElasticsearchVersion": "2.3",
    "ElasticsearchClusterConfig": {
        "InstanceType": ""m3.medium.elasticsearch"",
        "InstanceCount": 6,
        "DedicatedMasterEnabled": "true",
        "DedicatedMasterCount": 3,
        "DedicatedMasterType": "m3.medium.elasticsearch",
        "ZoneAwarenessEnabled": "true"
    }
}
```

Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 30b03e92-536f-11e5-9cd2-b36dbf43d89e
Content-Type: application/json
Content-Length: 645
Date: Sat, 05 Sep 2015 01:41:15 GMT
{
    "DomainStatus": {
        "ARN": "arn:aws:es:us-east-1:123456789012:domain/streaming-logs",
        "AccessPolicies": "",
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "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": "m3.medium.elasticsearch",
            "InstanceCount": 6,
            "InstanceType": "m3.medium.elasticsearch",
            "ZoneAwarenessEnabled": true
        },
        "ElasticsearchVersion": "2.3",
        "Endpoint": null,
        "Processing": true,
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
        }
    }
}
```

DeleteElasticsearchDomain

Deletes an Amazon ES domain and all of its data. A domain cannot be recovered after it is deleted. Use the `DELETE` HTTP method with this operation.

Syntax

```
DELETE /2015-01-01/es/domain/<DOMAIN_NAME>
```

Request Parameters

Parameter	Data Type	Required	Description
`DomainName`	`DomainName` (p. 99)	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. 104)	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 /2015-01-01/es/domain/weblogs
```

Response

```
HTTP/1.1 200 OK
{
    "DomainStatus": {
        "ElasticsearchClusterConfig": {
            "DedicatedMasterEnabled": true,
            "InstanceCount": 6,
            "ZoneAwarenessEnabled": true,
            "DedicatedMasterType": "m3.medium.elasticsearch",
            "InstanceType": "m3.medium.elasticsearch",
            "DedicatedMasterCount": 3
        },
        "ElasticsearchVersion": "2.3",
        "Endpoint": "search-weblogs-6qjchdvprdi2v2d2ga5apylts4.us-
east-1.es.a9.com",
        "Created": true,
        "Deleted": true,
        "DomainName": "test-logs",
        "EBSOptions": {
            "EBSEnabled": false
        },
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
```

```
        },
        "DomainId": "123456789012/weblogs",
        "AccessPolicies": "",
        "Processing": true,
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "ARN": "arn:aws:es:us-east-1:123456789012:domain/weblogs"
    }
}
```

DescribeElasticsearchDomain

Describes the domain configuration for the specified Amazon ES domain, including the domain ID, domain service endpoint, and domain ARN. Use the HTTP `GET` method with this operation.

Syntax

```
GET /2015-01-01/es/domain/<DOMAIN_NAME>
```

Request Parameters

Parameter	Data Type	Required?	Description
`DomainName`	`DomainName` (p. 99)	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. 101)	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_DOMAIN>.amazonaws.com/2015-01-01/es/domain/streaming-logs
```

Response

```
HTTP/1.1 200 OK
{
    "DomainStatus": {
        "ElasticsearchClusterConfig": {
            "DedicatedMasterEnabled": true,
            "InstanceCount": 3,
            "ZoneAwarenessEnabled": false,
            "DedicatedMasterType": "m3.medium.elasticsearch",
            "InstanceType": "m3.medium.elasticsearch",
            "DedicatedMasterCount": 3
        },
        "ElasticsearchVersion": "2.3",
        "Endpoint": "search-streaming-logs-okga24ftzsbz2a2hzhsqw73jpy.us-
east-1.es.a9.com",
        "Created": true,
        "Deleted": false,
        "DomainName": "streaming-logs",
        "EBSOptions": {
            "EBSEnabled": false
        },
        "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"
    }
}
```

DescribeElasticsearchDomains

Describes the domain configuration for up to five specified Amazon ES domains. Information includes the domain ID, domain service endpoint, and domain ARN. Use the HTTP `POST` method with this operation.

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. 104)	Yes	Array of Amazon ES domains in the following format: {"DomainNames": ["<Domain_Name>","<Domain_Name>"...]

Response Elements

Field	Data Type	Description
DomainStatusList	ElasticsearchDomainStatusList (p. 132)	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_DOMAIN>.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": "1.5",
            "Endpoint": "search-streaming-logs-okga24ftzsbz2a2hzhsqw73jpy.us-
east-1.es.a9.com",
            "Created": true,
            "Deleted": false,
            "DomainName": "streaming-logs",
            "EBSOptions": {
                "EBSEnabled": false
            },
            "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": "1.5",
            "Endpoint": "search-logs-p5st2kbt77diuihoqi6omd7jiu.us-
east-1.es.a9.com",
            "Created": true,
            "Deleted": false,
            "DomainName": "logs",
            "EBSOptions": {
                "Iops": 4000,
                "VolumeSize": 512,
                "VolumeType": "io1",
                "EBSEnabled": true
            },
            "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"
        }
    ]
}
```

ListDomainNames

Displays the names of all Amazon ES domains owned by the current user.

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. 99)	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"},
{"DomainName":"movies"}]}%
```

DescribeElasticsearchDomainConfig

Displays the configuration of an Amazon ES domain. Use the HTTP `GET` method with this operation.

Syntax

```
GET /2015-01-01/es/domain/<DOMAIN_NAME>/config
```

Request Parameters

Parameter	Data Type	Required	Description
`DomainName`	`DomainName` (p. 99)	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. 100)	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": {
        "ElasticsearchClusterConfig": {
            "Status": {
                "PendingDeletion": false,
                "State": "Active",
                "CreationDate": 1436913638.995,
                "UpdateVersion": 15,
                "UpdateDate": 1437078577.8
            },
            "Options": {
                "DedicatedMasterEnabled": true,
                "InstanceCount": 1,
                "ZoneAwarenessEnabled": false,
                "DedicatedMasterType": "search.m3.medium",
                "InstanceType": "search.m3.xlarge",
                "DedicatedMasterCount": 3
            }
        },
        "AdvancedOptions": {
            "Status": {
                "PendingDeletion": false,
                "State": "Active",
                "CreationDate": 1436913638.995,
                "UpdateVersion": 6,
                "UpdateDate": 1436914324.278
            },
            "Options": {
                "rest.action.multi.allow_explicit_index": "true"
            }
        },
        "EBSOptions": {
            "Status": {
                "PendingDeletion": false,
                "State": "Active",
                "CreationDate": 1436913638.995,
                "UpdateVersion": 15,
                "UpdateDate": 1437078577.8
            },
            "Options": {
                "Iops": 4000,
                "VolumeSize": 512,
                "VolumeType": "io1",
                "EBSEnabled": true
            }
        },
        "AccessPolicies": {
            "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/*\"}]}"
        },
        "SnapshotOptions": {
            "Status": {
                "PendingDeletion": false,
                "State": "Active",
                "CreationDate": 1436913638.995,
                "UpdateVersion": 6,
                "UpdateDate": 1436914324.278
            },
            "Options": {
                "AutomatedSnapshotStartHour": 0
            }
        }
    }
}
```

ListTags

Displays all of the tags for an Amazon ES domain. Use the `GET` HTTP method with this operation.

Syntax

```
GET /2015-01-01/tags?arn=<DOMAIN_ARN>
```

Request Parameters

Parameter	Data Type	Required	Description
`ARN`	`ARN` (p. 98)	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. 105)	List of resource tags. For more information, see Tagging Amazon Elasticsearch Service Domains (p. 54).

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=aws:es:us-
east-1:123456789012:domain/logs/
```

Response

```
HTTP/1.1 200 OK
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "MacOS"
        },
        {
            "Key": "project",
            "Value": "trident"
        }
    ]
}
```

RemoveTags

Removes the specified resource tags from an Amazon ES domain. Use the `POST` HTTP method with this operation.

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. 98)	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. 105)	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. Use the POST HTTP method with this operation.

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>"
    },
    "AccessPolicies": "<ACCESS_POLICY_DOCUMENT>",
    "SnapshotOptions": {
        "AutomatedSnapshotStartHour": <START_HOUR>,
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "<TRUE|FALSE>",
            "indices.fielddata.cache.size": "<PERCENTAGE_OF_HEAP>"
        }
    }
```

Request Parameters

This operation does not use HTTP request parameters.

Request Body

Parameter	Data Type	Require	Description
`DomainName`	`DomainName` (p. 99)	Yes	Name of the Amazon ES domain for which you want to update the configuration.
`ElasticsearchClusterConfig`	`ElasticsearchClusterConfig` (p. 100)	No	Desired changes to the cluster configuration, such as the instance type and number of EC2 instances.
`EBSOptions`	`EBSOptions` (p. 99)	No	Type and size of EBS volumes attached to data nodes.
`SnapshotOptions`	`SnapshotOptionsStatus` (p. 104)	No	Hour during which the service takes an automated daily snapshot of the indices in the Amazon ES domain.
`AdvancedOptions`	`AdvancedOptionsStatus` (p. 97)	No	Key-value pairs to specify the following advanced options: • `rest.action.multi.allow_explicit_index` - 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 information, see Configuring Access Policies (p. 26).

Parameter	Data Type	Require	Description
			• `indices.fielddata.cache.size` - Specifies the percentage of Java heap space that is allocated to field data. By default, this setting is unbounded. For more information, see Field Data in the Elasticsearch documentation.
`AccessPolicies`	String	No	Specifies the access policies for the Amazon ES domain. For more information, see Configuring Access Policies (p. 26).

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": "",
            "Status": {
```

```
            "CreationDate": 1441929121.098,
            "PendingDeletion": false,
            "State": "Active",
            "UpdateDate": 1441929121.098,
            "UpdateVersion": 22
        }
    },
    "AdvancedOptions": {
        "Options": {
            "indices.fielddata.cache.size": "",
            "rest.action.multi.allow_explicit_index": "true"
        },
        "Status": {
            "CreationDate": 1441230557.921,
            "PendingDeletion": false,
            "State": "Active",
            "UpdateDate": 1441231536.923,
            "UpdateVersion": 19
        }
    },
    "EBSOptions": {
        "Options": {
            "EBSEnabled": false,
            "Iops": null,
            "VolumeSize": null,
            "VolumeType": null
        },
        "Status": {
            "CreationDate": 1440706652.618,
            "PendingDeletion": false,
            "State": "Active",
            "UpdateDate": 1440707401.259,
            "UpdateVersion": 6
        }
    },
    "ElasticsearchClusterConfig": {
        "Options": {
            "DedicatedMasterCount": 3,
            "DedicatedMasterEnabled": true,
            "DedicatedMasterType": "m3.medium.elasticsearch",
            "InstanceCount": 3,
            "InstanceType": "m3.medium.elasticsearch",
            "ZoneAwarenessEnabled": false
        },
        "Status": {
            "CreationDate": 1440706652.618,
            "PendingDeletion": false,
            "State": "Active",
            "UpdateDate": 1441231536.923,
            "UpdateVersion": 19
        }
    },
    "SnapshotOptions": {
        "Options": {
            "AutomatedSnapshotStartHour": 3
        },
        "Status": {
            "CreationDate": 1440706652.618,
            "PendingDeletion": false,
```

```
                "State": "Active",
                "UpdateDate": 1441929119.716,
                "UpdateVersion": 22
            }
        }
    }
}
```

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 string 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 information, see URL-based Access Control. For more information about access policies for sub-resources, see Configuring Access Policies (p. 26).
`indices.fielddata.cache.size`	Key-value string 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. For more information, see Field Data in the Elasticsearch documentation.

AdvancedOptionsStatus

Status of an update to advanced configuration options for an Amazon ES domain.

Field	Data Type	Description
Options	`AdvancedOptions` (p. 97)	Key-value string pairs to specify advanced Elasticsearch configuration options
Status	`OptionStatus` (p. 103)	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.

CreateElasticsearchDomainRequest

Container for the parameters required by the `CreateElasticsearchDomain` service operation.

Field	Data Type	Description
`DomainName`	`DomainName` (p. 99)	Name of the Amazon ES domain to create.
`ElasticsearchClusterConfig`	`ElasticsearchClusterConfig` (p. 100)	Container for the cluster configuration of an Amazon ES domain.
`EBSOptions`	`EBSOptions` (p. 99)	Container for the parameters required to enable EBS-based storage for an Amazon ES domain. For more information, see Configuring EBS-based Storage (p. 23).
`AccessPolicies`	String	IAM policy document specifying the access policies for the new Amazon ES domain. For more information, see Configuring Access Policies (p. 26).
`SnapshotOptions`	`SnapshotOptionsStatus` (p. 104)	Container for parameters required to configure automated snapshots of domain indices. For more information, see Configuring Snapshots (p. 31).
`AdvancedOptions`	`AdvancedOptionsStatus` (p. 97)	Key-value pairs to specify the following advanced options: • `rest.action.multi.allow_explicit_index` • `indices.fielddata.cache.size`

Field	Data Type	Description
		For more information, see Configuring Advanced Options (p. 32).

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. 23).

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.

Field	Data Type	Description
`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. 34).
`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. 36).

ElasticsearchDomainConfig

Container for the configuration of an Amazon ES domain.

Field	Data Type	Description
`ElasticsearchClusterConfig`	`ElasticsearchClusterConfig` (p. 305)	Container for the cluster configuration of an Amazon ES domain.
`EBSOptions`	`EBSOptions` (p. 99)	Container for EBS options configured for an Amazon ES domain.

Field	Data Type	Description
`AccessPolicies`	String	Specifies the access policies for the Amazon ES domain. For more information, see Configuring Access Policies (p. 26).
`SnapshotOptions`	`SnapshotOptionsStatus` (p. 104)	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. 31).
`AdvancedOptions`	`AdvancedOptionsStatus` (p. 97)	Key-value pairs to specify the following advanced options: • `rest.action.multi.allow_explicit_inde` - 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 information, see Configuring Access Policies (p. 26). • `indices.fielddata.cache.size` - Specifies the percentage of Java heap space that is allocated to field data. By default, this setting is unbounded. For more information, see Field Data in the Elasticsearch documentation.

ElasticsearchDomainStatus

Container for the contents of a `DomainStatus` data structure.

Field	Data Type	Description
`DomainID`	`DomainID` (p. 99)	Unique identifier for an Amazon ES domain.
`DomainName`	`DomainName` (p. 99)	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

Field	Data Type	Description
		and must be between 3 and 28 characters. Valid characters are a-z (lowercase only), 0-9, and – (hyphen).
`ElasticsearchVersion`	String	Version of Elasticsearch. Currently, Amazon ES supports 1.5 and 2.3.
`ARN`	`ARN` (p. 98)	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. 104)	Domain-specific endpoint used to submit index, search, and data upload requests to an Amazon ES domain.
`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.
`ElasticsearchClusterConfig`	`ElasticsearchClusterConfig` (p. 100)	Container for the cluster configuration of an Amazon ES domain.
`EBSOptions`	`EBSOptions` (p. 99)	Container for the parameters required to enable EBS-based storage for an Amazon ES domain. For more information, see Configuring EBS-based Storage (p. 23).

Field	Data Type	Description
`AccessPolicies`	String	IAM policy document specifying the access policies for the new Amazon ES domain. For more information, see Configuring Access Policies (p. 26).
`SnapshotOptions`	`SnapshotOptions` (p. 104)	Container for parameters required to configure the time of daily automated snapshots of Amazon ES domain indices.
`AdvancedOptions`	`AdvancedOptions` (p. 97)	Key-value string pairs to specify advanced Elasticsearch configuration options.

ElasticsearchDomainStatusList

List that contains the status of each specified Amazon ES domain.

Field	Data Type	Description
`DomainStatusList`	`ElasticsearchDomainStatus` (p. 101)	List that contains the status of each specified Amazon ES domain.

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`	Timestamp	Date and time when the Amazon ES domain was created
`UpdateDate`	Timestamp	Date and time when the Amazon ES domain was updated

Field	Data Type	Description
`UpdateVersion`	Integer	Whole number that specifies the latest version for the entity
`State`	`OptionState` (p. 103)	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. 104)	Container for parameters required to configure the time of daily automated snapshots of indices in an Amazon ES domain
`Status`	`OptionStatus` (p. 103)	Status of an update to snapshot options for an Amazon ES domain

Tag

Field	Data Type	Description
`Key`	`TagKey` (p. 105)	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. 54).
`Value`	`TagValue` (p. 105)	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. 105)	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.

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.

Exception	Description
`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.
`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 describe limits for Amazon Elasticsearch Service (Amazon ES).

Cluster Resource	Limit
Maximum Number of Instances (Instance Count) per Cluster	20 (except for t2 instance types, which have a maximum of 10).
Maximum Number of Dedicated Master Nodes	5 **Note** You can use a T2 instance type as a dedicated master only if the instance count is 10 or fewer.
Reserved Storage Space per Cluster	The service reserves 10% to 20% of storage space on each instance for segment merges, logs, and other internal operations.
Swapping Memory to Disk	The service disables memory swapping for all supported instance types except the following: • t2.micro.elasticsearch • t2.small.elasticsearch • t2.medium.elasticsearch

EBS Resource	Description
Minimum and Maximum Size of EBS Volumes	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. Minimum and maximum EBS volume sizes by instance type: Minimum: 10 GB; Maximum: 35 GB for each data node

EBS Resource	Description
	Minimum: 10 GB; Maximum: 100 GB Minimum: 10 GB; Maximum: 512 GB Minimum and maximum EBS volume sizes by volume type: Minimum: 10 GB; Maximum: 512 GB for each data node Minimum: 35 GB; Maximum: 512 GB for each data node Minimum: 10 GB; Maximum: 100 GB for each data node For example, the maximum size for a gp2 EBS volume type attached to a t2.small.elasticsearch instance type is 35 GB, not 512 GB, because the maximum EBS volume size for this instance type is 35 GB. If you want a larger gp2 EBS volume, you must attach it to a larger instance type. Calculate the total amount of EBS-based storage for the Amazon ES domain using the following formula: (number of data nodes) * (EBS volume size).

Amazon ES Resource	Limit
`bootstrap.mlockall`	The service enables `bootstrap.mlockall` in `elasticsearch.yml`, which locks JVM memory and prevents the operating system from swapping it to disk, for all supported instance types except the following: • t2.micro.elasticsearch • t2.small.elasticsearch • t2.medium.elasticsearch
Custom Plugins	The service does not support custom plugins. However, the service does support several standard plugins, including Kibana, ICU, and Kuromoji.
`ES_HEAP_SIZE` Environment Variable	The service limits Java processes to a heap size of 32 GB.
Scripting Module	The service does not support dynamic scripting.

Logstash Resource	Limit
Output Plugins	The service supports two output plugins: the standard elasticsearch plugin and the logstash-output-amazon-es plugin provided by Amazon ES, which signs and exports Logstash events to Amazon ES.

Region Resource	Limit
sa-east-1 and us-east-2	The sa-east-1 and us-east-2 regions do not support i2 and r3 instance types.

Region Resource	Limit
us-east-2, ap-northeast-2, and ap-south-1	The us-east-2, ap-northeast-2, and ap-south-1 regions support m4 instance types instead of m3 instance types.

Service Request Resource	Limit
Maximum Size of HTTP Request Payloads	Each supported instance type has a maximum supported payload for HTTP requests: • 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.medium.elasticsearch: 10 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 • r3.large.elasticsearch: 100 MB • r3.xlarge.elasticsearch: 100 MB • r3.2xlarge.elasticsearch: 100 MB • r3.4xlarge.elasticsearch: 100 MB • r3.8xlarge.elasticsearch: 100 MB • i2.xlarge.elasticsearch: 100 MB • i2.2xlarge.elasticsearch: 100 MB
Parameters in HTTP POST Requests	The service ignores parameters passed in URLs for HTTP POST requests signed with Signature Version 4.
TCP Transport	The service supports HTTP on port 80, but does not support TCP transport.

Document History for Amazon Elasticsearch Service

This topic describes important changes to the documentation for Amazon Elasticsearch Service (Amazon ES).

Relevant Dates to this History:

- **Current product version—**2015-01-01
- **Latest product release—**2016-7-29
- **Latest documentation update—**10 December 2016

Change	Description	Release Date
US East (Ohio) Support	Amazon Elasticsearch Service added support for the following region: US East (Ohio), `us-east-2`. For a list of regions supported by Amazon Elasticsearch Service, see AWS Regions and Endpoints in the *AWS General Reference*.	October 17, 2016
New Performance Metric	Amazon Elasticsearch Service added a performance metric, ClusterUsedSpace.	July 29, 2016
Elasticsearch 2.3 Support	Amazon Elasticsearch Service added support for Elasticsearch version 2.3.	July 27, 2016
Asia Pacific (Mumbai) Support	Amazon Elasticsearch Service added support for the following region: Asia Pacific (Mumbai), `ap-south-1`. For a list of regions supported by Amazon Elasticsearch Service, see AWS Regions and Endpoints in the *AWS General Reference*.	June 27, 2016
More Instances per Cluster	Amazon Elasticsearch Service increased the maximum number of instances (instance count) per cluster from 10 to 20.	May 18, 2016

Change	Description	Release Date
Asia Pacific (Seoul) Support	Amazon Elasticsearch Service added support for the following region: Asia Pacific (Seoul), `ap-northeast-2`. For a list of regions supported by Amazon Elasticsearch Service, see AWS Regions and Endpoints in the *AWS General Reference*.	January 28, 2016
Amazon Elasticsearch Service	Initial release.	October 1, 2015

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.