Amazon ElastiCache

User Guide

API Version 2011-07-15

Amazon ElastiCache: User Guide

Welcome

This is the *Amazon ElastiCache User Guide*. This guide picks up where the Amazon ElastiCache Getting Started Guide leaves off, and helps you understand the components that the Amazon ElastiCache provides and how to use them. The guide shows you how to access Amazon ElastiCache with a web-based GUI, with command line tools, and programmatically through the Amazon ElastiCache API.

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale distributed in-memory cache environments in the cloud. It provides a high performance, resizable, and cost-effective in-memory cache, while removing the complexity associated with deploying and managing a distributed cache environment.

How Do I...?

How Do I?	Relevant Sections
Get a general product overview and information about pricing	Amazon ElastiCache product page
Get a quick hands-on introduction to Amazon ElastiCache	Amazon ElastiCache Getting Started Guide
Learn about Amazon ElastiCache key terminology and concepts	Amazon ElastiCache Terminology and Concepts (p. 5)
How to get started with the command line tools.	Setting up the Command Line Tools (p. 19)
Get started using the Query API for Amazon ElastiCache	Using the Amazon ElastiCache API (p. 23)
Find available libraries for programmatically accessing Amazon ElastiCache	Available Libraries (p. 29)
Get detailed information about how to use Amazon ElastiCache components and features, with instructions for several different interfaces	Using Amazon ElastiCache (p. 31)

How Do I?	Relevant Sections
Learn how to set up a Cache Cluster	Creating a Cache Cluster (p. 33)

Introduction to Amazon ElastiCache

Topics

What is Amazon ElastiCache?

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale distributed in-memory cache environments in the cloud. It provides a high performance, resizable, and cost-effective in-memory cache, while removing the complexity associated with deploying and managing a distributed cache environment. Amazon ElastiCache is protocol-compliant with Memcached, so the code, applications, and most popular tools that you use today with your existing Memcached environments will work seamlessly with the service.

Advantages of Amazon ElastiCache

Amazon ElastiCache provides the following advantages:

- **Simple to Deploy—** Amazon ElastiCache makes it very easy to deploy a distributed in-memory cache environment. Use the AWS Management Console or simple API calls to access the capabilities of a production-ready Cache Cluster in minutes, without worrying about infrastructure provisioning or installing and maintaining cache software.
- **Managed—**Amazon ElastiCache automates time-consuming management tasks, such as patch management, failure detection and recovery, allowing you to pursue higher value application development.
- **Compatible—**With Amazon ElastiCache, you get native access to a Memcached environment. This facilitates compatibility with your existing tools and applications. In addition, Amazon ElastiCache gives you optional control over which supported Memcached Version powers your Cache Cluster via Cache Cluster Version Management.
- **Elastic—**With a simple API call or a few clicks of the AWS Management Console, you can add or delete Cache Nodes to your Cache Cluster to meet your business needs and application load.

- **Reliable—**Amazon ElastiCache has multiple features that enhance reliability for critical production deployments, including automatic detection of Cache Node failures and recovery. Amazon ElastiCache runs on the same highly reliable infrastructure used by other Amazon Web Services.
- **Designed for use with other AWS products—**Amazon ElastiCache works in conjunction with other Amazon Web Services (such as Amazon EC2, Amazon CloudWatch and Amazon SNS) to provide a secure, high-performance and managed in-memory cache. For example, an application running in Amazon EC2 can securely access an Amazon ElastiCache Cluster in the same Region with very low latency.
- **Secure—**Amazon ElastiCache provides web service interfaces to configure firewall settings that control network access to your Cache Cluster.
- **Inexpensive—**You pay for memory/compute capacity by the hour, with no long-term commitments. This frees you from the costs and complexities of planning, purchasing, and maintaining hardware, and transforms what are commonly large fixed costs into much smaller variable costs.

Amazon ElastiCache Terminology and Concepts

Topics

This chapter introduces you to Amazon ElastiCache terminology and concepts. Many of the concepts introduced in this chapter are explored in greater depth in later chapters.

Cache Cluster

A Cache Cluster is a collection of one or more Cache Nodes, each running an instance of the Memcached service. Most of your operations will be performed at the Cache Cluster level. A Cache Cluster can be set up with a specific number of Cache Nodes and a Cache Parameter Group that controls the properties for each Cache Node. All Cache Nodes within a Cache Cluster are designed to be of the same Node Type and have the same parameter and security group settings.

Cache Node

A Cache Node is the smallest building block of an Amazon ElastiCache deployment. It is a fixed-size chunk of secure, network-attached RAM. Each Cache Node runs an instance of the Memcached service, and has its own DNS name and port. Multiple types of Cache Nodes are supported, each with varying amounts of associated memory.

Cache Cluster Identifier

The Cache Cluster Identifier is a customer-supplied identifier for a Cache Cluster. This identifier specifies a particular Cache Cluster when interacting with the Amazon ElastiCache API and commands. The Cache Cluster Identifier must be unique for that customer in an AWS region.

Cache Security Groups

Amazon ElastiCache allows you to control access to your Cache Clusters using Cache Security Groups. A Cache Security Group acts like a firewall, controlling network access to your Cache Cluster. By default, network access is turned off to your Cache Clusters. If you want your applications to access your Cache Cluster, you must explicitly enable access from hosts in specific EC2 security groups. Once ingress rules are configured, the same rules apply to all Cache Clusters associated with that CacheSecurityGroup.

To allow network access to your Cache Cluster, create a Cache Security Group and use the AuthorizeCacheSecurityGroupIngress API or CLI command to authorize the desired EC2 security group (which in turn specifies the EC2 instances allowed). The Cache Security Group can be associated with your Cache Cluster at the time of creation, or using a ModifyCacheCluster command.

Important

IP-range based access control is currently not enabled for Cache Clusters. All clients to a Cache Cluster must be within the EC2 network, and authorized via security groups as described above.

For more information on working with Cache Security Groups, go to Working with Cache Security Groups (p. 59).

Cache Parameter Groups

Amazon ElastiCache allows you to control the runtime parameters of your Cache Nodes using Cache Parameter Groups. A CacheParameterGroup represents a combination of specific values for each of the parameters passed to Memcached during startup, therefore determining how the Memcached process on each Cache Node will behave at runtime. The parameter values on a specific CacheParameterGroup apply to all Cache Nodes on all Cache Clusters associated with that group.

To see the list of supported parameters, their default values, and which ones can be modified, use the DescribeEngineDefaultParameters API.

To create, describe, modify, and delete parameter groups, you can use the CreateCacheParameterGroup, DescribeCacheParameterGroup (for summary information about all parameter groups under the account), DescribeCacheParameters (for details on the values of each parameter in a group), ModifyCacheParameterGroup and DeleteCacheParameter Group APIs. In addition, you can also reset all the values in a parameter group to the default values using ResetCacheParameterGroup.

You can associate a Cache Cluster with a parameter group when you create it, and all associated parameters will be applied to the Cache Cluster. You can change specific values in a parameter group at any time (using the **ModifyCacheParameterGroup**) API, and change the parameter group associated with a Cache Cluster at any time (using the **ModifyCacheCluster** API). Note that such changes will not be applied immediately to the running Cache Clusters – a reboot of each Cache Node in the Cache Cluster is required before the updated parameters take effect. A reboot of one or more Cache Nodes in a Cache Cluster can be initiated using the **RebootCacheCluster** API.

For more information on working with Cache Parameter Groups, go to Working with Cache Parameter Groups (p. 48).

Available Cache Parameters

This section provides details about the parameters that Amazon ElastiCache makes available for your Cache Clusters.

Available Memcached Parameters

The following table shows the Memcached 1.4.5 parameters that Amazon ElastiCache supports.

Name	Default	Type	Modifiable	Description
backlog_queue_limit	1024	integer	No	The backlog queue limit.
binding_protocol	auto	string	No	The binding protocol.
cas_disabled	0	boolean	Yes	If supplied, check and set (CAS) operations will be disabled, and items stored will consume 8 bytes less than with CAS enabled.
chunk_size	48	integer	Yes	The minimum amount of space to allocate for the smallest item's key, value, and flags, in bytes.
chunk_size_growth_factor	1.25	float	Yes	The growth factor controlling the size of each successive memcached chunk; each chunk will be chunk_size_growth_factor times larger than the previous chunk.
error_on_memory_exhausted	0	Boolean	Yes	If supplied, when there is no more memory to store items, memcached will return an error rather than evicting items.
large_memory_pages	0	Boolean	Yes	Try to use large memory pages.
lock_down_paged_memory	0	Boolean	Yes	Lock down all paged memory.
max_item_size	1048576	integer	Yes	The size of the largest item storable in the cache, in bytes.
max_simultaneous_connections	10000	integer	No	The maximum number of simultaneous connections.

Name	Default	Type	Modifiable	Description
maximize_core_file_limit	0	boolean	No	Maximize the core file limit.
memcached_connections_overhead	100	integer	Yes	The amount of memory to be reserved for memcached connections and other miscellaneous overhead. For important information about understanding and tuning this parameter, see the following section
requests_per_event	20	integer	No	The maximum number of requests per event, limits the number of requests processed for a given connection to prevent starvation.

Understanding and Tuning Memcached Connection Overhead

On each Cache Node, the memory made available for storing cache items is the total available memory on that Cache Node (which is stored in the `max_cache_memory` parameter) minus the memory used for connections and other overhead (which is stored in the `memcached_connections_overhead` parameter). As an example, a Cache Node of type `cache.m1.small` has a total of 1300MB `max_cache_memory`. With the default `memcached_connections_overhead` value of 100MB, the Memcached process will be given 1200MB for storing cache items.

We have configured default values for the `memcached_connections_overhead` parameter to satisfy most use cases. However, the required amount of allocation for connection overhead can vary depending on multiple factors, including request rate, payload size, and the number of connections.

You can change the value of the `memcached_connections_overheard` to better suit the needs of your application. For example, increasing the value of the `memcached_connections_overhead` parameter will reduce the amount of memory available for storing cache items and provide a larger buffer for connection overhead, while decreasing the value of the `memcached_connections_overhead` parameter will give you more memory to store cache items, but can increase your risk of swap usage and degraded performance. If you observe swap usage and degraded performance, try increasing the value of the `memcached_connections_overhead` parameter.

Available Cache Node Type-Specific Parameters

While most parameters have a single value that affects the Cache Node, some parameters have different values depending on the Cache Node Type used. These are called out separately in the tables below.

The following table shows the default values for the `max_cache_memory` and `num_threads` parameters for each Cache Node Type.

Note

The `max_cache_memory` and `num_threads` parameters are not modifiable.

Cache Node Type	max_cache_memory	num_threads
cache.c1.xlarge	6600	8
cache.m1.small	1300	1

Cache Node Type	max_cache_memory	num_threads
cache.m1.large	7100	2
cache.m1.xlarge	14600	4
cache.m2.xlarge	16600	2
cache.m2.2xlarge	33800	2
cache.m2.4xlarge	68000	8

Configuring Cache Clients

Amazon ElastiCache is protocol-compliant with Memcached, so the code, applications, and most popular tools that you use today with your existing Memcached environments will work seamlessly with the service. This section discusses specific considerations for using Amazon ElastiCache.

DNS Names and Underlying IP

Memcached clients maintain a server list containing the addresses and ports of the servers holding the cache data. When using Amazon ElastiCache, the DescribeCacheClusters API (or the elasticache-describe-cache-clusters command line utility) returns a fully qualified DNS entry and port number that can be used for the server list.

Important

It is important that client applications are configured to frequently resolve DNS names of cache nodes when they attempt to connect to a cache node endpoint.

Amazon ElastiCache will ensure that the DNS name of a Cache Node is unchanged when Cache Nodes are recovered in case of failure; however, the underlying IP address of the cache node can change.

Most Memcached client libraries support persistent cache node connections by default, and we recommend using persistent cache node connections when using Amazon ElastiCache. Client-side DNS caching can occur in multiple places, including the Memcached client library, the language runtime, or the client operating system. You should review your application configuration at each layer to ensure that you are frequently resolving IP addresses for your cache nodes.

Choosing a Cache Node Type and the Number of Cache Nodes

One of the advantages of Amazon ElastiCache is that you can easily select and change the amount of memory and compute capacity available in your Cache Cluster. This topic will assist you in the process of choosing the initial capacity of your Cache Cluster and show you how to change the capacity in existing Cache Clusters.

The total memory capacity of your Cache Cluster is calculated by multiplying the number of Cache Nodes in the Cluster by the capacity of each Node. The capacity of each Cache Node is based on the Cache Node type.

The number of Cache Nodes in the Cache Cluster is a key factor in the availability of your Cache Cluster. The failure of a single Cache Node can have an impact on the availability of your application and the load on your backend database while Amazon ElastiCache provisions a replacement for the failed Cache Node. The scale of this availability impact can be reduced by spreading your memory and compute capacity over a larger number of Cache Nodes, each with smaller capacity, rather than a fewer number of high capacity nodes.

In a scenario where you want to have 20GB of cache memory, you can set it up in one of the following ways:

- Use 15 `cache.m1.small` Cache Nodes with 1.3 GB of memory each = 19.5 GB
- Use 3 `cache.m1.large` Cache Nodes with 7.1 GB of memory each = 21.3 GB
- Use 3 `cache.c1.xlarge` Cache Nodes with 6.6 GB of memory each = 19.8 GB

These options provide you with similar memory capacity, but different computational capacity for your Cache Cluster.

Note

Some of the available memory on each Cache Node will be used for connection overhead. For more information, see the section Understanding and Tuning Memcached Connection Overhead (p. 8).

If you're unsure about how much capacity you need, we recommend starting with one `cache.m1.small` Cache Node Type and monitoring the memory usage, CPU utilization and Cache Hit Rate with the ElastiCache metrics that are published to Amazon CloudWatch.

If your Cache Cluster does not have the desired hit rate, you can easily add more nodes, thereby increasing the total available memory in your Cache Cluster. You will need to obtain an updated endpoint list from the ElastiCache CLI, API or AWS Management Console, and configure your clients to use the additional node(s).

If your Cache Cluster turns out to be bound by CPU but has sufficient hit rate, then try setting up a new Cluster with a different Cache Node Type.

Amazon ElastiCache supports adding or removing Cache Nodes from an existing Cache Cluster using the AWS Management Console, the API, and the command line tools, allowing you to increase both memory and compute capacity of the Cluster at any time.

Note

Amazon ElastiCache does not currently support dynamically changing the Cache Node Type for a Cache Cluster after it has been created. If you wish to change the Node Type of a Cache Cluster, you will need to set up a new Cache Cluster with the desired Node Type, and migrate your application to that Cache Cluster.

For more information about how to scale Cache Clusters using the API, the command line tools, or the AWS Management Console, see Scaling a Cache Cluster (p. 38).

Cache Engine Version Management

You can control if and when the Memcached protocol-compliant software powering your Cache Cluster is upgraded to new versions that are supported by Amazon ElastiCache. This enables you to maintain

compatibility with specific Memcached versions, test new versions with your application before deploying in production, and perform version upgrades on your own terms and timelines.

Since major version upgrades involve some compatibility risk, they will not occur automatically and must be initiated by you. Automatic patching works only for minor upgrades; for example, from version 1.4.5 to 1.4.6. Automatic patching will not work for major upgrades (for example, from version 1.4.x to 1.5.x). You can manually initiate major version upgrades to your Cache Cluster by modifying the Cache Cluster and specifying a new engine version.

Note

While Cache Engine Version Management functionality is intended to give you as much control as possible over how patching occurs, Amazon ElastiCache reserves the right to patch your Cache Cluster on your behalf in the event of a critical security vulnerability in the system or cache software.

Automatic patching will occur during the scheduled maintenance window for your Cache Clusters, and will be announced on the Amazon ElastiCache discussion forum in advance. Automatic version upgrades are enabled by default on new Cache Clusters; if you wish to turn off automatic version upgrades, you can do so by selecting the “Auto Minor Version Upgrade” for your Cache Cluster to “No”.

Regions and Availability Zones

Amazon cloud computing resources are housed in highly available data center facilities. To provide additional scalability and reliability, these data center facilities are located in several different physical locations. These locations are categorized by *Regions* and *Availability Zones*.

Regions are large and widely dispersed into separate geographic locations. Availability Zones are distinct locations within a Region that are engineered to be isolated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region.

Important

Each Region is completely independent. Any Amazon ElastiCache activity you initiate (e.g. creating Cache Clusters) runs only in your current default Region. The default Region can be changed by setting the EC2_REGION environment variable, or be overridden by using the `--url` parameter with the command line interface. See *Common Options for API Tools* for more information.

To create or work with a Cache Cluster in a specific Region, use the corresponding regional service endpoint.

Amazon ElastiCache supports the endpoints listed in the following table.

Region	Endpoint
US East (Northern Virginia) Region	https://elasticache.us-east-1.amazonaws.com
US West (Northern California) Region	https://elasticache.us-west-1.amazonaws.com
US West (Oregon) Region	https://elasticache.us-west-2.amazonaws.com

Region	Endpoint
EU (Ireland) Region	https://elasticache.eu-west-1.amazonaws.com
Asia Pacific (Singapore) Region	https://elasticache.ap-southeast-1.amazonaws.com
Asia Pacific (Tokyo) Region	https://elasticache.ap-northeast-1.amazonaws.com
South America (Sao Paulo) Region	https://elasticache.sa-east-1.amazonaws.com

If you do not explicitly specify an endpoint, the US-East (Northern Virginia) Region endpoint is the default.

Maintenance Window

Every cache cluster has a weekly maintenance window during which any system changes are applied. If you don't specify a preferred maintenance window when you create the Cache Cluster, Amazon ElastiCache assigns a 30-minute maintenance window on a randomly selected day of the week.

The 30-minute maintenance window is selected at random from an 8-hour block of time per region. The following table lists the time blocks for each Region from which the default maintenance windows are assigned.

Region	Time Block
US East (Northern Virginia) Region	03:00-11:00 UTC
US West (Northern California) Region	06:00-14:00 UTC
US West (Oregon) Region	06:00-14:00 UTC
EU (Ireland) Region	22:00-06:00 UTC
Asia Pacific (Singapore) Region	14:00-22:00 UTC
Asia Pacific (Tokyo) Region	17:00-03:00 UTC
South America (Sao Paulo) Region	01:00-09:00 UTC

The maintenance window should fall at the time of lowest usage and thus might need modification from time to time. You can specify a time range of up to 24 hours in duration during which any maintenance activities you have requested should occur. For example, if you enable Auto Minor Version Upgrades for your Cache Cluster, non-critical Memcache software updates are applied during this time. Such updates occur infrequently (generally once every few months) and will be announced on the AWS ElastiCache forum two weeks prior to being applied. Any deferred or pending Cache Cluster modifications you have requested would also occur during this time.

For more information on how to adjust the preferred maintenance window for your Cache Clusters, see Adjusting the Preferred Maintenance Window (p. 42).

CloudWatch Metrics with Amazon ElastiCache

Amazon ElastiCache provides metrics that enable you to monitor your Cache Clusters. You can access these metrics through Amazon CloudWatch. For more information on Amazon CloudWatch, go to the Amazon CloudWatch documentation.

Metric Dimensions

All Amazon ElastiCache metrics use the "AWS/ElastiCache" namespace and provide metrics for a single dimension, the *CacheNodeId*, which is the automatically-generated identifier for each Cache Node in the Cache Cluster. You can find out what these values are for your Cache Nodes using the `DescribeCacheClusters` API or **elasticache-describe-cache-clusters** command line utility.

Each metric is published under a single set of dimensions. When retrieving metrics, you must supply both the CacheClusterId and CacheNodeId dimensions.

Available Metrics

Amazon ElastiCache provides both host-level metrics (for example, CPU usage) and Memcached-specific metrics (i.e. number of gets). These metrics are measured and published for each Cache node in 60-second intervals.

The following table lists Memcached-specific metrics provided by Amazon ElastiCache at the Cache Node level.

Metric	Description	Unit
CPUUtilization	The percentage of CPU utilization.	Percent
SwapUsage	The amount of swap used on the host.	Bytes
FreeableMemory	The amount of free memory available on the host.	Bytes
NetworkBytesIn	The number of bytes the host has read from the network.	Bytes
NetworkBytesOut	The number of bytes the host has written to the network.	Bytes

The following table lists the cache node-level metrics provided by Amazon ElastiCache that are derived from the Memcached stats command.

Note

For complete documentation of the Memcached stats command, go to https://github.com/memcached/memcached/blob/master/doc/protocol.txt.

Metric	Description	Unit
BytesUsedForCacheItems	The number of bytes used to store cache items.	Bytes
BytesReadIntoMemcached	The number of bytes the cache has written to client connections.	Bytes

Metric	Description	Unit
CasBadval	The number of CAS (check and set) requests the cache has received where the Cas value did not match the Cas value stored.	Count
CasHits	The number of Cas requests the cache has received where the requested key was found and the Cas value matched.	Count
CasMisses	The number of Cas requests the cache has received where the key requested was not found.	Count
CmdFlush	The number of flush commands the cache has received.	Count
CmdGet	The number of get commands the cache has received.	Count
CmdSet	The number of set commands the cache has received.	Count
CPUUtilization	The percentage of CPU utilization.	Percent
CurrConnections	A count of the number of connections connected to the cache at an instant in time. Note that due to the design of Memcached, this will always return a minimum count of 10.	Count
CurrItems	A count of the number of items currently stored in the cache.	Count
DecrHits	The number of decrement requests the cache has received where the requested key was found.	Count
DecrMisses	The number of decrement requests the cache has received where the requested key was not found.	Count
DeleteHits	The number of delete requests the cache has received where the requested key was found.	Count
DeleteMisses	The number of delete requests the cache has received where the requested key was not found.	Count
Evictions	The number of non-expired items the cache evicted to allow space for new writes.	Count
GetHits	The number of get requests the cache has received where the key requested was found.	Count
GetMisses	The number of get requests the cache has received where the key requested was not found.	Count
IncrHits	The number of increment requests the cache has received where the key requested was found.	Count
IncrMisses	The number of increment requests the cache has received where the key requested was not found.	Count

Metric	Description	Unit
Reclaimed	The number of expired items the cache evicted to allow space for new writes.	Count

The following table describes the available calculated cache level metrics.

Metric	Description	Unit
NewConnections	The number of new connections the cache has received. This is derived from the memcached total_connections statistic by recording the change in total_connections across a period of time. This will always be at least 1, due to a connection reserved for a Amazon ElastiCache.	Count
NewItems	The number of new items the cache has stored. This is derived from the memcached total_items statistic by recording the change in total_items across a period of time.	Count
UnusedMemory	The amount of unused memory the cache can use to store items. This is derived from the memcached statistics limit_maxbytes and bytes by subtracting bytes from limit_maxbytes.	Bytes

Choosing Metric Statistics and Periods

While Amazon CloudWatch will allow you to choose any statistic and period for each metric, not all combinations will be useful. For example, the Average, Minimum, and Maximum statistics for CPUUtilization are useful, but the Sum statistic is not.

All Amazon ElastiCache samples are published for a 60 second duration for each individual Cache Node. For any 60 second period, a Cache Node metric will only contain a single sample.

For further information on how to retrieve metrics for your Cache Nodes, see Viewing Cache Cluster and Cache Node Metrics (p. 66).

Using Amazon Simple Notification Service (SNS) with Amazon ElastiCache

Amazon ElastiCache can publish messages using the Amazon Simple Notification Service (SNS) when significant events happen on a Cache Cluster. This feature can be used to refresh the server-lists on client machines connected to individual Cache Node endpoints of a Cache Cluster.

Note

For more information on Amazon Simple Notification Service (SNS), including information on pricing and links to the Amazon SNS documentation, go to the Amazon SNS product page.

Notifications are published to a specified Amazon SNS *topic*. Only one topic can be configured for Amazon ElastiCache notifications, and the AWS account that owns the Amazon SNS topic must be the same account that owns the Cache Cluster on which notifications are enabled.

The following Amazon ElastiCache events trigger Amazon SNS notifications:

Event Name	Description
ElastiCache:CacheClusterProvisioningComplete	Indicates that the provisioning of a Cache Cluster is completed, and the Cache Nodes in the Cache Cluster are ready to use.
ElastiCache:DeleteCacheClusterComplete	Indicates that the deletion of a Cache Cluster and all associated Cache Nodes has completed.
ElastiCache:CacheClusterSecurityGroupModified	Indicates that one of the following events has occurred: • The list of Cache Security Groups authorized for the Cache Cluster has been modified. • One or more new EC2 security groups have been authorized on any of the Cache Security Groups associated with the Cache Cluster. • One or more EC2 security groups have been revoked from any of the Cache Security Groups associated with the Cache Cluster.
ElastiCache:AddCacheNodesComplete	Indicates that Cache Nodes have been added to the Cache Cluster and are ready for use.
ElastiCache:RemoveCacheNodesComplete	Indicates that Cache Nodes have been removed from the Cache Cluster.
ElastiCache:CacheNodeReplacementComplete	Indicates that Amazon ElastiCache has detected that the host running a Cache Node is degraded or unreachable and has completed replacing the Cache Node. **Note** The DNS entry for the replaced Cache Node is not changed. In most instances, you do not need to refresh the server-list for your clients when this event occurs. However, some cache client libraries may stop using the Cache Node even after Amazon ElastiCache has replaced the Cache Node; in this case, the application should refresh the server-list when this event occurs.

For more information on using Amazon SNS notifications with Amazon ElastiCache, see Using Amazon SNS Notifications with Amazon ElastiCache (p. 44).

AWS Identity and Access Management

Amazon ElastiCache integrates with AWS Identity and Access Management (IAM), a service that lets your organization do the following:

- Create users and groups under your organization's AWS account
- Easily share your AWS account resources between the users in the account
- Assign unique security credentials to each user
- Granularly control users access to services and resources
- Get a single AWS bill for all users under the AWS account

For example, you can use IAM with Amazon ElastiCache to control which Users in your AWS Account can create or modify Cache Clusters for your AWS Account.

For general information about IAM, go to:

- Identity and Access Management (IAM)
- AWS Identity and Access Management Getting Started Guide
- Using AWS Identity and Access Management

For specific information about how you can control User access to Amazon ElastiCache, go to Integrating with Other AWS Products in *Using AWS Identity and Access Management*.

For more information on using IAM with Amazon ElastiCache, go to Controlling User Access to Your AWS Account (p. 23).

Signing up for Amazon ElastiCache

To use Amazon ElastiCache, you must first sign up for the service. After you sign up for the service, you can get your user credentials and start using Amazon ElastiCache.

To use Amazon ElastiCache, you need an AWS account. If you don't already have one, you'll be prompted to create one when you sign up for Amazon ElastiCache.

To sign up for Amazon ElastiCache

1. Go to http://aws.amazon.com/elasticache and click **Sign Up for Amazon ElastiCache**.
2. Follow the on-screen instructions.

Setting up the Command Line Tools

Topics

This section describes the prerequisites for running the command line tools, where to get the command line tools, how to set up the tools and their environment, and includes a series of common examples of tool usage.

Prerequisites

This document assumes that you can work in a Linux/UNIX or Windows environment. The Amazon ElastiCache command line tools also work correctly on Mac OS X (which resembles the Linux and UNIX command environment), but no specific Mac OS X instructions are included in this guide.

As a convention, all command line text is prefixed with a generic **`PROMPT>`** command line prompt. The actual command line prompt on your machine is likely to be different. We also use **`$`** to indicate a Linux/UNIX specific command and **`C:\>`** for a Windows specific command. The example output resulting from the command is shown immediately thereafter without any prefix.

The Java Runtime Environment

The command line tools used in this guide require Java version 5 or later to run. Either a JRE or JDK installation is acceptable. To view and download JREs for a range of platforms, including Linux/UNIX and Windows, go to Java SE Downloads.

Setting the Java Home Variable

The command line tools depend on an environment variable (`JAVA_HOME`) to locate the Java Runtime. This environment variable should be set to the full path of the directory that contains a sub directory

named `bin` which in turn contains the executable `java` (on Linux and UNIX) or `java.exe` (on Windows) executable.

To set the Java Home variable

1. Set the Java Home variable.

   - On Linux and UNIX, enter the following command:

     ```
     $ export JAVA_HOME=<PATH>
     ```

   - On Windows, enter the following command:

     ```
     C:\> set JAVA_HOME=<PATH>
     ```

2. Confirm the path setting by running **`$JAVA_HOME/bin/java -version`** and checking the output.

   - On Linux/UNIX, you will see output similar to the following:

     ```
     $ $JAVA_HOME/bin/java -version
     java version "1.6.0_23"
     Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
     Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
     ```

   - On Windows, you will see output similar to the following:

     ```
     C:\> %JAVA_HOME%\bin\java -version
     java version "1.6.0_23"
     Java(TM) SE Runtime Environment (build 1.6.0_23-b05)
     Java HotSpot(TM) Client VM (build 19.0-b09, mixed mode, sharing)
     ```

Getting the Command Line Tools

The command line tools are available as a ZIP file on the Amazon ElastiCache Developer Tools web site. These tools are written in Java, and include shell scripts for Windows 2000/XP/Vista/Windows 7, Linux/UNIX, and Mac OSX. The ZIP file is self-contained and no installation is required; simply download the zip file and unzip it to a directory on your local machine.

Setting Up the Tools

The command line tools depend on an environment variable (AWS_ELASTICACHE_HOME) to locate supporting libraries. You need to set this environment variable before you can use the tools. Set it to the path of the directory you unzipped the command line tools into. This directory is named

ElastiCacheCli-A.B.nnnn (A, B and n are version/release numbers), and contains sub-directories named bin and lib.

To set the AWS_ELASTICACHE_HOME environment variable

- Open a command line window and enter one of the following commands to set the AWS_ELASTICACHE_HOME environment variable.
  - On Linux and UNIX, enter the following command:

```
$ export AWS_ELASTICACHE_HOME=<path-to-tools>
```

  - On Windows, enter the following command:

```
C:\> set AWS_ELASTICACHE_HOME=<path-to-tools>
```

To make the tools easier to use, we recommend that you add the tools' BIN directory to your system PATH. The rest of this guide assumes that the BIN directory is in your system path.

To add the tools' BIN directory to your system path

- Enter the following commands to add the tools' BIN directory to your system PATH.
  - On Linux and UNIX, enter the following command:

```
$ export PATH=$PATH:$AWS_ELASTICACHE_HOME/bin
```

  - On Windows, enter the following command:

```
C:\> set PATH=%PATH%;%AWS_ELASTICACHE_HOME%\bin
```

Note

The Windows environment variables are reset when you close the command window. You might want to set them permanently. Consult the documentation for your version of Windows for more information.

Note

Paths that contain a space must be wrapped in double quotes, for example:

"C:\Program Files\Java"

Providing Credentials for the Tools

The command line tools need the AWS Access Key and Secret Access Key provided with your AWS account. You can get them using the command line or from a credential file located on your local system.

The deployment includes a template file ${AWS_ELASTICACHE_HOME}/credential-file-path.template that you need to edit with your information. Following are the contents of the template file:

```
AWSAccessKeyId=<Write your AWS access ID>
AWSSecretKey=<Write your AWS secret key>
```

Important

On UNIX, limit permissions to the owner of the credential file:

```
$ chmod 600 <the file created above>
```

With the credentials file setup, you'll need to set the AWS_CREDENTIAL_FILE environment variable so that the Amazon ElastiCache tools can find your information.

To set the AWS_CREDENTIAL_FILE environment variable

1. Set the environment variable:

   - On Linux and UNIX, update the variable using the following command:

     ```
     $ export AWS_CREDENTIAL_FILE=<the file created above>
     ```

   - On Windows, set the variable using the following command:

     ```
     C:\> set AWS_CREDENTIAL_FILE=<the file created above>
     ```

2. Check that your setup works properly, run the following command:

   ```
   elasticache --help
   ```

   You should see the usage page for all Amazon ElastiCache commands.

Using the Amazon ElastiCache API

This section provides task-oriented descriptions of how to use and implement Amazon ElastiCache operations. For a complete description of Amazon ElastiCache operations, see the Amazon ElastiCache API Reference.

Topics

Controlling User Access to Your AWS Account

Topics

Amazon ElastiCache allows you to control access to your Cache Clusters using Cache Security Groups. A Cache Security Group acts like a firewall controlling network access to your Cache Cluster.

Important

Amazon ElastiCache uses Cache Security Groups to control who has access to specific Amazon ElastiCache Cache Clusters. There's no way in the IAM system to allow or deny access to a specific Cache Cluster.

For more information about using Security Groups with Amazon ElastiCache, refer to the Amazon ElastiCache User Guide.

Amazon ElastiCache Group Security and IAM

Using IAM with Amazon ElastiCache doesn't change how you use Amazon ElastiCache Cache Security Groups to grant access to Cache Clusters. However, you can use IAM policies to specify which Amazon ElastiCache actions a User in your AWS Account can use with Amazon ElastiCache resources in general. Because you can't specify a particular Cache Cluster in the policy, you must specify * as the resource to indicate all Cache Clusters in the AWS Account.

Example

You could create a policy that gives the Developers group permission to use only these APIs: `CreateCacheCluster`, `DescribeCacheClusters`, `ModifyCacheCluster`, `RebootCacheCluster`, `DeleteCacheCluster`, `DescribeEvents`. They could then use those APIs with any Cache Cluster that belongs to your AWS Account.

For examples of IAM policies that cover Amazon ElastiCache actions, see Example Policies for Amazon ElastiCache (p. 25).

No Amazon ElastiCache ARNs

Because you can't specify a particular Amazon ElastiCache resource in an IAM policy, Amazon ElastiCache has no ARNs. When writing a policy to control access to Amazon ElastiCache actions, you use * as the resource. For more information about ARNs, go to ARNs in the AWS Identity and Access Management (IAM) documentation.

Amazon ElastiCache Actions

In an IAM policy, you can specify any and all actions that Amazon ElastiCache offers. Each action name must be prefixed with the lowercase string `elastiCache:`. For example: `elastiCache:ModifyCacheCluster`, `elastiCache:DescribeCacheCluster`, or `elastiCache:*` (for all Amazon ElastiCache actions). For a list of the actions, refer to the Query API action names in the Amazon ElastiCache API Reference.

Amazon ElastiCache Keys

Amazon ElastiCache implements the following policy keys, but no others. For more information about policy keys, go to Condition in the in the AWS Identity and Access Management (IAM) documentation.

AWS-Wide Policy Keys

- `aws:CurrentTime` (for date/time conditions)
- `aws:EpochTime` (the date in epoch or UNIX time, for use with date/time conditions)
- `aws:SecureTransport` (Boolean representing whether the request was sent using SSL)
- `aws:SourceIp` (the requester's IP address, for use with IP address conditions)
- `aws:UserAgent` (information about the requester's client application, for use with string conditions)

If you use `aws:SourceIp`, and the request comes from an Amazon EC2 instance, we evaluate the instance's public IP address to determine if access is allowed.

For services that use only SSL, such as Amazon RDS and Amazon Route 53, the `aws:SecureTransport` key has no meaning.

The key names are case insensitive. For example, `aws:CurrentTime` is equivalent to `AWS:currenttime`.

Example Policies for Amazon ElastiCache

This section shows a few simple policies for controlling User access to Amazon ElastiCache.

Note

In the future, Amazon ElastiCache might add new actions that should logically be included in one of the following policies, based on the policy's stated goals.

Example 1: Allow a Network Admin group to only be able to access the APIs related to Amazon ElastiCache Security Groups

In this example, we create a policy that gives access to the relevant actions and attach it to the group. The resource is stated as "*", because you can't specify a particular Amazon ElastiCache resource in an IAM policy.

```
{
   "Statement":[{
      "Effect":"Allow",
      "Action":["elasticache:CreateCacheSecurityGroup",
                "elasticache:DeleteCacheSecurityGroup",
                "elasticache:DescribeCacheSecurityGroup",
                "elasticache:AuthorizeCacheSecurityGroupIngress",
                "elasticache:RevokeCacheSecurityGroupIngress"],
      "Resource":"*"
      }
   ]
}
```

Example 2: Allow managers to only be able to list the current Amazon ElastiCache resources in the AWS Account

In this example, we create a policy that lets managers use the Amazon ElastiCache actions with `Describe` in the name.

```
{
   "Statement":[{
      "Effect":"Allow",
      "Action":"elasticache:Describe*",
      "Resource":"*"
      }
   ]
}
```

Example 3: Allow a system administrator to access a select set of Amazon ElastiCache actions

In this example, we create a policy that gives access to the relevant actions for system administrators and attach it to the group. As with the other examples, the resource is stated as "*", because you can't specify a particular Amazon ElastiCache resource in an IAM policy.

```
{
   "Statement":[{
      "Effect":"Allow",
      "Action":["elasticache:ModifyCacheCluster",
                "elasticache:RebootCacheCluster",
                "elasticache:DescribeCacheClusters",
                "elasticache:DescribeEvents",
                "elasticache:ModifyCacheParameterGroup",
                "elasticache:DescribeCacheParameterGroups",
                "elasticache:DescribeCacheParameters",
                "elasticache:ResetCacheParameterGroup",
                "elasticache:DescribeEngineDefaultParameters"],
      "Resource":"*"
      }
   ]
}
```

Making API Requests

Endpoints

For information about this product's regions and endpoints, go to Regions and Endpoints in the Amazon Web Services General Reference.

If no endpoint is explicitly specified, the US-East (Northern Virginia) Region endpoint is used by default.

Using the Query API

Topics

Query Parameters

HTTP Query-based requests are HTTP requests that use the HTTP verb GET or POST and a Query parameter named *`Action`*.

Each Query request must include some common parameters to handle authentication and selection of an action.

Some operations take lists of parameters. These lists are specified using the *`param.n`* notation. Values of *`n`* are integers starting from 1.

Query Request Authentication

You can only send Query requests over HTTPS and you must include a signature in every Query request. This section describes how to create the signature. The method described in the following procedure is known as *signature version 2*.

The following are the basic steps used to authenticate requests to AWS. This assumes you are registered with AWS and have an Access Key ID and Secret Access Key.

Query Authentication Process

1	The sender constructs a request to AWS.
2	The sender calculates the request signature, a Keyed-Hashing for Message Authentication Code (HMAC) with a SHA-1 hash function, as defined in the next section of this topic.
3	The sender of the request sends the request data, the signature, and Access Key ID (the key-identifier of the Secret Access Key used) to AWS.
4	AWS uses the Access Key ID to look up the Secret Access Key.
5	AWS generates a signature from the request data and the Secret Access Key using the same algorithm used to calculate the signature in the request.
6	If the signatures match, the request is considered to be authentic. If the comparison fails, the request is discarded, and AWS returns an error response.

Note

If a request contains a `Timestamp` parameter, the signature calculated for the request expires 15 minutes after its value. If a request contains an `Expires` parameter, the signature expires at the time specified by the `Expires` parameter.

Calculating the request signature

1. Create the canonicalized query string that you need later in this procedure:

   a. Sort the UTF-8 query string components by parameter name with natural byte ordering. The parameters can come from the GET URI or from the POST body (when Content-Type is application/x-www-form-urlencoded).

   b. URL encode the parameter name and values according to the following rules:

      i. Do not URL encode any of the unreserved characters that RFC 3986 defines. These unreserved characters are A-Z, a-z, 0-9, hyphen (-), underscore (_), period (.), and tilde (~).

      ii. Percent encode all other characters with %XY, where X and Y are hex characters 0-9 and uppercase A-F.

      iii. Percent encode extended UTF-8 characters in the form %XY%ZA....

      iv. Percent encode the space character as %20 (and not +, as common encoding schemes do).

   c. Separate the encoded parameter names from their encoded values with the equals sign (=) (ASCII character 61), even if the parameter value is empty.

d. Separate the name-value pairs with an ampersand (&) (ASCII code 38).

2. Create the string to sign according to the following pseudo-grammar (the "\n" represents an ASCII newline).

```
StringToSign = HTTPVerb + "\n" +
ValueOfHostHeaderInLowercase + "\n" +
HTTPRequestURI + "\n" +
CanonicalizedQueryString <from the preceding step>
```

   The HTTPRequestURI component is the HTTP absolute path component of the URI up to, but not including, the query string. If the HTTPRequestURI is empty, use a forward slash (/).

3. Calculate an RFC 2104-compliant HMAC with the string you just created, your Secret Access Key as the key, and SHA256 or SHA1 as the hash algorithm.

   For more information, go to http://www.rfc.net/rfc2104.html.

4. Convert the resulting value to base64.

5. Include the value as the value of the *Signature* parameter in the request.

For example, the following is an example request (linebreaks added for clarity).

```
https://elasticache.amazonaws.com/
?Action=DescribeCacheClusters
 &CacheClusterIdentifier=mycachecluster
 &Version=2011-07-15
 &Timestamp=2011-07-19T17%3A09%3A03.726Z
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &AWSAccessKeyId=<Your AWS Access Key ID>
```

For the preceding Query string, you would calculate the HMAC signature over the following string.

```
GET\n
elasticache.amazonaws.com\n
AWSAccessKeyId=<Your AWS Access Key ID>
&Action=DescribeCacheClusters
&CacheClusterIdentifier=mycachecluster
&Timestamp=2011-07-19T17%3A09%3A03.726Z
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Version=2011-07-15
```

The result is the following signed request.

```
https://elasticache.amazonaws.com/
```

```
?Action=DescribeCacheClusters
&CacheClusterIdentifier=mycachecluster
&Version=2011-07-15
&Timestamp=2011-07-19T17%3A09%3A03.726Z
&Signature=<URLEncode(Base64Encode(Signature))>
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&AWSAccessKeyId=<Your AWS Access Key ID>
```

Available Libraries

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of the Query API. These libraries provide basic functions (not included in the APIs), such as request authentication, request retries, and error handling so that it is easier to get started. Libraries and resources are available for the following languages:

- Java
- PHP
- Python
- Ruby
- Windows and .NET

For libraries and sample code in all languages, go to Sample Code & Libraries.

Troubleshooting Applications

Topics

Amazon ElastiCache provides specific and descriptive errors to help you troubleshoot problems while interacting with the Amazon ElastiCache API.

Retrieving Errors

Typically, you want your application to check whether a request generated an error before you spend any time processing results. The easiest way to find out if an error occurred is to look for an *Error* node in the response from the Amazon ElastiCache API.

XPath syntax provides a simple way to search for the presence of an *Error* node, as well as an easy way to retrieve the error code and message. The following code snippet uses Perl and the XML::XPath module to determine if an error occurred during a request. If an error occurred, the code prints the first error code and message in the response.

```
use XML::XPath;
               my $xp = XML::XPath->new(xml =>$response);
               if ( $xp->find("//Error") )
```

```
                {print "There was an error processing your request:\n", " Error
code: ",
                 $xp->findvalue("//Error[1]/Code"), "\n", " ",
                 $xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Troubleshooting Tips

We recommend the following processes to diagnose and resolve problems with the Amazon ElastiCache API.

- Verify that Amazon ElastiCache is running correctly
  To do this, simply open a browser window and submit a Query request to the Amazon ElastiCache service (such as https://elasticache.amazonaws.com). A MissingAuthenticationTokenException or 500 Internal Server Error confirms that the service is available and responding to requests.
- Check the structure of your request
  Each Amazon ElastiCache operation has a reference page in the *Amazon ElastiCache API Reference*. Double-check that you are using parameters correctly. In order to give you ideas regarding what might be wrong, look at the sample requests or user scenarios to see if those examples are doing similar operations.
- Check the forum
  Amazon ElastiCache has a development community forum where you can search for solutions to problems others have experienced along the way. To view the forum, go to
  http://developer.amazonwebservices.com/connect/forum.jspa?forumID=60

Using Amazon ElastiCache

Topics

This section covers the Amazon ElastiCache operations you are most likely to use, and provides procedural instruction and examples.

Creating and Modifying Cache Clusters

The following scenarios cover basic operations on Cache Clusters.

Topics

Creating a Cache Cluster

When you create a new Cache Cluster, you need to name it, and choose the number and type of Cache Nodes.

The following procedures show you how to create a Cache Cluster called *mycachecluster*, with a *Large* Cache Node Type.

Note

For information on accessing a Cache Cluster, refer to the Amazon ElastiCache Getting Started Guide.

AWS Management Console

To launch a Cache Cluster

1. From the Amazon ElastiCache Console, click **Launch Cache Cluster** to start the Launch Cache Cluster Wizard.

   The wizard opens on the **Cache Cluster Details** page.
2. Specify the name of your Cache Cluster by typing it into the **Name** text box.
3. Select the amount of memory you want for the Cache Nodes by selecting a Node Type from the **Node Type** drop-down list box.
4. Select the number of Cache Nodes you want in your Cache Cluster by typing a number into the **Number of Nodes** text box.
5. You can change the port that your Cache Nodes will use to accept connections by typing it into the **Cache Port** text box.
6. Leave the rest of the options on the **Cache Cluster Details** page at their default values, and click the **Continue** button.
7. After you click the **Continue** button, the **Additional Configuration** page opens.
8. The **Additional Configuration** page is where you can specify a Cache Security Group, Cache Parameter Group, and Maintenance Window for you Cache Cluster. For this example, accept the default values, and then click the **Continue** button.

   After you click the **Continue** button, the **Review** panel appears.
9. Review the options for your Cache Cluster:

   - If you need to correct any options, click the **Back** button to return to previous panels and make corrections.
   - If all your options are entered correctly, click the **Launch** button to launch your new Cache Cluster.

     After you click the **Launch** button, a message displays stating that your Cache Cluster is being created.

     This can take a few minutes to complete.
10. Click the **Close** button.

    After you click the **Close** Instance button, the **My Cache Clusters** panel appears. Your Cache Cluster appears in the list on this page with the **creating** status until your Cache Cluster is created and ready for use.

CLI

To create a Cache Cluster

- Use the command `elasticache-create-cache-cluster` to create a Cache Cluster.

```
PROMPT>elasticache-create-cache-cluster simcoprod42 -n 3 -c cache.m1.large
 -e memcached -sg default
```

This command should produce output similar to the following:

```
CACHECLUSTER  simcoprod42  cache.m1.large  memcached  creating  3  1.4.5
 SECGROUP  default  active
 PARAMGRP  default.memcached1.4  in-sync
```

API

To create a Cache Cluster

- Call `CreateCacheCluster` with the following parameters:
  - *`CacheClusterId`* = `simcoprod42`
  - *`CacheNodeType`* = `cache.m1.large`
  - *`Engine`* = `memcached`
  - *`NumCacheNodes`* = `3`

Example

```
https://elasticache.us-east-1.amazonaws.com/
   ?CacheNodeType=cache.m1.large
   &CacheSecurityGroupNames.member.1=default
   &NumCacheNodes=3
   &CacheClusterId=simcoprod42
   &Engine=memcached
   &Version=2011-07-15
   &Action=CreateCacheCluster
   &SignatureVersion=2
   &SignatureMethod=HmacSHA256
   &Timestamp=2011-07-27T02%3A14%3A07.821Z
   &AWSAccessKeyId=<AWS Access Key ID>
   &Signature=<Signature>
```

Related Topics

- Cache Cluster (p. 5)
- Scaling a Cache Cluster (p. 38)
- Deleting a Cache Cluster (p. 36)

Deleting a Cache Cluster

As soon as your Cache Cluster becomes available, you're billed for each hour or partial hour that you keep the Cache Cluster running (even if the Cache Cluster is idle). Once you've decided that you no longer need the Cache Cluster, you can delete it. Deleting a Cache Cluster requires you to identify the Cache Cluster you want to remove.

The following procedures show you how to delete a Cache Cluster.

AWS Management Console

To delete a Cache Cluster

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. In the **My Cache Clusters** list, click the **Delete** link next to the Cache Cluster you wish to delete.

   The **Delete Cache Cluster** confirmation dialog box appears.
3. Click the **Yes, Delete** button.

CLI

To delete a Cache Cluster

- Use the command `elasticache-delete-cache-cluster` to delete a Cache Cluster.

```
PROMPT>elasticache-delete-cache-cluster mycachecluster mycachecluster --
force
```

This command will produce output similar to the following:

```
CACHECLUSTER  mycachecluster  2011-07-27T23:05:34.725Z  cache.m1.small
memcache
d  deleting  3  us-east-1d  1.4.5
   SECGROUP  default  active
   PARAMGRP  default.memcached1.4  in-sync
```

API

To delete a Cache Cluster

- Call `DeleteCacheCluster` with the following parameters:
  - *`CacheClusterId`* = `mycachecluster`

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?Action=DeleteCacheCluster
 &CacheClusterId=mycachecluster
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-27T22%3A20%3A46.297Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Related Topics

- Creating a Cache Cluster

Scaling a Cache Cluster

You can scale a Cache Cluster by adding or removing Cache Nodes to accommodate changes in your application's capacity requirements.

The following procedures show you how to scale a Cache Cluster by adding and removing Cache Nodes.

Scaling Up a Cache CLuster

AWS Management Console

To scale up a Cache Cluster

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. In the **My Cache Clusters** list, click the Cache Cluster you want to scale up in the **My Cache Clusters** list. The detail panel appears.
3. Click the **Nodes** tab. The list of cache nodes for the cache cluster appears.
4. Click the **Add Node(s)** button at the top of the list. The **Add Node(s)** dialog box appears.
5. Type the number of nodes you want to add into the **Number of nodes to add** text box.
6. Select the **Apply Immediately** check box.
7. Click the **Yes, Add** button.

   After a few moments, the new cache nodes will show up in the Nodes list with a status of "Creating".

CLI

To scale up a Cache Cluster by adding Cache Nodes

- Use the command `elasticache-modify-cache-cluster` with the following parameters:

```
PROMPT>elasticache-modify-cache-cluster simcoprod02 --num-cache-nodes 5 --
apply-immediately
```

This command produces output similar to the following:

```
CACHECLUSTER  simcoprod02  2011-07-06T23:34:09.756Z  cache.m1.large  memcached
modifying  3  us-east-1b  1.4.5  5
   SECGROUP  default  active
   PARAMGRP  default.memcached1.4  in-sync
```

API

To scale up a Cache Cluster

- Call `ModifyCacheCluster` with the following parameters:
  - *`CacheClusterId`* = `simcoprod02`

- *NumCacheNodes* = 5
- *ApplyImmediately* = true

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?Action=ModifyCacheCluster
 &ApplyImmediately=true
 &NumCacheNodes=5
 &CacheClusterId=simcoprod02
 &Version=2011-07-15
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-14T23%3A34%3A45.892Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Scaling Down a Cache Cluster

AWS Management Console

The following procedures show how you can scale down a Cache Cluster.

To scale down a Cache Cluster

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. In the **My Cache Clusters** list, click the Cache Cluster you want to scale down in the **My Cache Clusters** list. The detail panel appears.
3. Click the **Nodes** tab. The list of cache nodes for the cache cluster appears.
4. In the Nodes list, click the check box next to the Cache Nodes you want to remove from the Cache Cluster.
5. Click the **Remove Node(s)** button.
   The **Remove Node(s)** confirmation dialog box appears.
6. Click the **Apply Immediately** checkbox.
7. Click the **Yes, Remove** button.

   After a few moments, the new cache nodes will show up in the Nodes list with a status of "deleting".

CLI

To scale down a Cache Cluster by removing Cache Nodes

1. Use the command `elasticache-describe-cache-cluster` to display a list of cache nodes for a Cache Cluster, as in the following example, and note the identifiers of the cache nodes you wish to remove.

```
PROMPT>elasticache-describe-cache-clusters simcoprod02 -sn
```

This command produces output similar to the following:

```
CACHECLUSTER  simcoprod02  2011-07-06T23:34:09.756Z  cache.m1.large  memcached
available  5  us-east-1b  1.4.5
   SECGROUP  default  active
   PARAMGRP  default.memcached1.4  in-sync
   CACHENODE  0001  2011-07-14T23:39:51.273Z  available  simcoprod02.m2st2p.f
sw4.use1qa.cache.amazonaws.com  11211  in-sync
   CACHENODE  0002  2011-07-14T23:39:51.276Z  available  simcoprod02.m2st2p.f
sw7.use1qa.cache.amazonaws.com  11211  in-sync
   CACHENODE  0003  2011-07-06T23:34:09.756Z  available  simcoprod02.m2st2p.f
swc.use1qa.cache.amazonaws.com  11211  in-sync
   CACHENODE  0004  2011-07-06T23:34:09.756Z  available  simcoprod02.m2st2p.f
swd.use1qa.cache.amazonaws.com  11211  in-sync
   CACHENODE  0005  2011-07-06T23:34:09.756Z  available  simcoprod02.m2st2p.f
swf.use1qa.cache.amazonaws.com  11211  in-sync
```

2. Use the command `elasticache-modify-cache-cluster` with a list of the cache clusters to remove, as in the following example.

```
PROMPT>elasticache-modify-cache-cluster simcoprod02 --num-cache-nodes 3 --
nodes-to-remove 0004,0005 --apply-immediately
```

This command produces output similar to the following:

```
CACHECLUSTER  simcoprod02  2011-07-06T23:34:09.756Z  cache.m1.large  memcached
modifying  3  us-east-1b  1.4.5  5
   SECGROUP  default  active
   PARAMGRP  default.memcached1.4  in-sync
```

API

To scale down a Cache Cluster

- Call `ModifyCacheCluster` with the Cache Cluster ID and a list of cache nodes to remove:
  - *`CacheClusterId`* = `simcoprod02`
  - *`NumCacheNodes`* = `3`
  - *`CacheClusterNodeIdsToRemove.member.1`* = `0004`
  - *`CacheClusterNodeIdsToRemove.member.2`* = `0005`
  - *`ApplyImmediately`* = `true`

Example

```
https://elasticache.us-east-1.amazonaws.com/
   ?Action=ModifyCacheCluster
   &CacheClusterId=simcoprod02
   &ApplyImmediately=true
   &CacheClusterNodeIdsToRemove.member.1=0004
   &CacheClusterNodeIdsToRemove.member.2=0005
   &NumCacheNodes=3
   &Version=2011-07-15
   &SignatureVersion=2
   &SignatureMethod=HmacSHA256
   &Timestamp=2011-07-16T00%3A10%3A55.608Z
   &AWSAccessKeyId=<AWS Access Key ID>
   &Signature=<Signature>
```

Related Topics

- Creating a Cache Cluster (p. 33)

Adjusting the Preferred Maintenance Window

Every cache cluster has a weekly maintenance window during which any system changes are applied. If you don't specify a preferred maintenance window when you create the Cache Cluster, Amazon ElastiCache assigns a one hour maintenance window on a randomly selected day of the week. The one hour maintenance window is selected at random from an 8-hour block of time per region.

The maintenance window should fall at the time of lowest usage and thus might need modification from time to time. You can specify the time range (UTC) during which any maintenance activities you have requested should occur. For example, if you enable Auto Minor Version Upgrades for your Cache Cluster, non-critical Memcache software updates are applied during this time. Such updates occur infrequently (generally once every few months) and will be announced on the AWS ElastiCache forum two weeks prior to being applied. Any deferred or pending Cache Cluster modifications you have requested would also occur during this time.

Adjusting the Maintenance Window

The following procedures show you how to adjust the preferred maintenance window for a Cache Cluster.

For the purpose of this example, we assume that the Cache Cluster named *mycachecluster* exists and has a preferred maintenance window of "Sun:05:00-Sun:06:00" UTC.

AWS Management Console

To adjust the preferred maintenance window

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. Click on the **Cache Clusters** link in the Navigation panel on the left side of the console display.
   The **My Cache Clusters** list appears.
3. Click the **Modify** link next to the Cache Cluster in the **My Cache Clusters** list.
   The **Modify Cache Cluster** window appears.
4. Select the starting and ending time in the **Maintenance Window** section of the **Modify Cache Cluster** dialog box.
5. Select the **Apply Immediately** check box.
6. Click the **Yes, Modify** button.

CLI

To adjust the preferred maintenance window

- Use the `elasticache-modify-cache-cluster` command with the following parameters:

```
PROMPT>elasticache-modify-cache-cluster simcoprod02 --preferred-maintenance-
window Tue:04:00-Tue:04:30
```

This command produces output similar to the following.

```
CACHECLUSTER  simcoprod02  2011-07-06T23:34:09.756Z  cache.m1.large  memcached
available  3  us-east-1b  1.4.5
```

```
SECGROUP  default  active
PARAMGRP  default.memcached1.4  in-sync
```

API

To adjust the preferred maintenance window

- Call `ModifyCacheCluster` with the following parameters:
  - *`CacheClusterId`* = `simcoprod02`
  - *`PreferredMaintenanceWindow`* = `Tue:04:00-Tue:04:30`

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?Action=ModifyCacheCluster
 &CacheClusterId=simcoprod02
 &PreferredMaintenanceWindow=Tue:04:00-Tue:04:30
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-24T17%3A48%3A21.746Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Related Topics

- Creating a Cache Cluster (p. 33)

Using Amazon SNS Notifications with Amazon ElastiCache

You can configure Amazon ElastiCache to send notifications for important Cache Cluster events using Amazon Simple Notification Service (Amazon SNS). In these examples, you will configure a Cache Cluster with the Amazon Resource Name (ARN) of an Amazon SNS topic to receive notifications.

Note

This topic assumes that you've signed up for Amazon SNS and have set up (and subscribed to) an Amazon SNS topic. For information on how to do this, please see the Amazon SNS Getting Started Guide.

Adding an Amazon SNS Topic

The following procedures show you how to add an Amazon SNS topic for a Cache Cluster.

Note

This process can also be used to modify the Amazon SNS topic.

AWS Management Console

To add an Amazon SNS notification topic ARN

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. In the **My Cache Clusters** list, click the **Modify** link next to the Cache Cluster to which you want to add an Amazon SNS topic ARN .

   The **Modify Cache Cluster** window appears.
3. Add the ARN of the Amazon SNS topic to the **Notification Topic ARN** text box.
4. Click the **Apply Immediately** check box.
5. Click the **Yes, Modify** button.

CLI

To add an Amazon SNS notification topic ARN

- Use the command `elasticache-modify-cache-cluster` with the following parameters:

```
PROMPT>elasticache-modify-cache-cluster simcoprod42 -t arn:aws:sns:us-east-
1:565419523791:ElastiCacheNotifications
```

This command produces output similar to the following:

```
CACHECLUSTER  simcoprod42  2011-07-26T01:21:46.607Z  cache.m1.large  memcached
available  3  us-east-1d  1.4.5
   SECGROUP  default  active
```

```
   PARAMGRP  default.memcached1.4  in-sync
  NOTIFICATION  arn:aws:sns:us-east-1:565419523791:ElastiCacheNotifications

   active
```

API

To add an Amazon SNS notification topic ARN

- Call `ModifyCacheCluster` with the following parameters:
  - *`CacheClusterId`* = `simcoprod42`
  - *`TopicArn`* = `arn:aws:sns:us-east-1:565419523791:ElastiCacheNotifications`

Example

```
https://elasticache.amazon.com/
 ?ApplyImmediately=false
 &CacheClusterId=simcoprod42
 &NotificationTopicArn=arn%3Aaws%3Asns%3Aus-east-1%3A565419523791%3AElastiC
acheNotifications
 &Version=2011-07-15
 &Action=ModifyCacheCluster
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-26T03%3A17%3A42.676Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Enabling and Disabling Amazon SNS Notifications

You can turn notifications on or off for a Cache Cluster. The following procedures show you how to disable Amazon SNS notifications.

AWS Management Console

To disable Amazon SNS notifications for a Cache Cluster

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. In the **My Cache Clusters** list, click the **Modify** link next to the Cache Cluster to which you want to add an Amazon SNS topic ARN .

   The **Modify Cache Cluster** window appears.
3. In the **Notification Topic Status** drop-down list box, select **`inactive`**.
4. Select the **Apply Immediately** check box.
5. Click the **Yes, Modify** button.

CLI

To disable Amazon SNS notifications for a Cache Cluster

- Use the command `elasticache-modify-cache-cluster` with the following parameters:

```
PROMPT>elasticache-modify-cache-cluster simcoprod42 -ts inactive
```

This command produces output similar to the following:

```
CACHECLUSTER  simcoprod42  2011-07-26T01:21:46.607Z  cache.m1.large  memcached
available  3  us-east-1d  1.4.5
   SECGROUP  default  active
   PARAMGRP  default.memcached1.4  in-sync
  NOTIFICATION  arn:aws:sns:us-east-1:565419523791:ElastiCacheNotifications

   inactive
```

API

To disable Amazon SNS notifications for a Cache Cluster

- Call `ModifyCacheCluster` with the following parameters:
  - *`CacheClusterId`* = `simcoprod42`
  - *`NotificationTopicStatus`* = `inactive`

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?ApplyImmediately=false
 &CacheClusterId=simcoprod42
 &NotificationTopicStatus=inactive
 &Version=2011-07-15
 &Action=ModifyCacheCluster
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-26T03%3A17%3A42.676Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Related Topics

- Using Amazon Simple Notification Service (SNS) with Amazon ElastiCache (p. 15)

Cache Parameter Groups

The following scenarios cover operations on Cache Parameter Groups.

Topics

Working with Cache Parameter Groups

A Cache Parameter Group is initially created with the default parameters for the cache engine used by the Cache Cluster. To provide custom values for any of the parameters, you must modify the group after creating it.

In this example, you create, list, modify, and examine Cache Parameter Groups.

Caution

Improperly setting parameters in a Cache Parameter Group can have unintended adverse effects, including degraded performance and system instability. Always exercise caution when you are modifying cache parameters or your Cache Parameter Group.

Note

Some cache engine parameters are constrained or disabled in the context of an ElastiCache Cache Cluster. For more information, please see Cache Parameter Groups (p. 6).

Creating a Cache Parameter Group

The following procedures show you how to create a new Cache Parameter Group.

AWS Management Console

To create a Cache Parameter Group

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. Click **Cache Parameter Groups** in the **Navigation** list on the left side of the window.
3. Click the **Create Cache Parameter Group** button.
   The **Create Cache Parameter Group** window appears.
4. The only cache parameter group family currently supported is `memcached1.4`. This is selected for you in the **Cache Parameter Group Family** drop-down list box.
5. Type the name of the new Cache Parameter Group in the **Cache Parameter Group** text box.
6. Type a description for the new Cache Parameter Group in the **Description** text box.
7. Click the **Yes, Create** button.

CLI

To create a Cache Parameter Group

- Use the command `elasticache-create-cache-parameter-group` with the following parameters:

```
PROMPT>elasticache-create-cache-parameter-group myCacheParameterGroup -fm
memecached1.4 -d "My new parameter group"
```

This command should produce output similar to the following:

```
CACHEPARAMETERGROUP  mycacheparametergroup  memcached1.4  My new parameter
 group
```

API

To create a Cache Parameter Group

- Call `CreateCacheParameterGroup` with the following parameters:
  - *`CacheParameterGroupName`* = `myCacheparamgroup`
  - *`Description`* = `"My new parameter group"`

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?Description=My%20new%20parameter%20group
 &CacheParameterGroupFamily=memcached1.4
 &CacheParameterGroupName=myCacheParameterGroup
 &Version=2011-07-15
 &Action=CreateCacheParameterGroup
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-16T01%3A13%3A41.537Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

This command should return a response similar to the following:

```
<CreateCacheParameterGroupResponse xmlns="http://elmo.amazonaws.com/doc/2011-
07-15/">
  <CreateCacheParameterGroupResult>
    <CacheParameterGroup>
     <CacheParameterGroupName>mycacheparametergroup</CacheParameterGroupName>

      <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>
      <Description>My new parameter group</Description>
    </CacheParameterGroup>
  </CreateCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>d8465952-af48-11e0-8d36-859edca6f4b8</RequestId>
  </ResponseMetadata>
</CreateCacheParameterGroupResponse>
```

Listing Available Cache Parameter Groups

You can see which Cache Parameter Groups you've created for your AWS account by listing them. The following procedures show you how to list all available Cache Parameter Groups.

Note

The `default.memcached1.4` parameter group family is automatically created the first time you create a Cache Cluster without specifying a custom Cache Parameter Group.

AWS Management Console

To list all available Cache Parameter Groups for an AWS account

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. Click **Cache Parameter Groups** in the **Navigation** list on the left side of the window.
   The available Cache Parameter Groups appears in the **My Cache Parameter Groups** list.

CLI

To list all available Cache Parameter Groups for an AWS account

- Use the command `elasticache-describe-cache-parameter-groups` to list all available Cache Parameter Groups for your AWS account:

```
PROMPT>elasticache-describe-cache-parameter-groups
```

This command should produce output similar to the following:

```
CACHEPARAMETERGROUP  default.memcached1.4  memcached1.4  Default parameter
 group
for memcached1.4
```

API

To list all available Cache Parameter Groups for an AWS account

- Call `DescribeCacheParameterGroups` with no parameters.

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?MaxRecords=100
 &Version=2011-07-15
 &Action=DescribeCacheParameterGroups
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-16T01%3A09%3A08.033Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

This command should return a response similar to the following:

```
<DescribeCacheParameterGroupsResponse xmlns="http://elmo.amazon
aws.com/doc/2011-07-15/">
  <DescribeCacheParameterGroupsResult>
    <CacheParameterGroups>
      <CacheParameterGroup>
        <CacheParameterGroupName>default.memcached1.4</CacheParameterGroup
Name>
        <CacheParameterGroupFamily>memcached1.4</CacheParameterGroupFamily>

        <Description>Default parameter group for memcached1.4</Description>

      </CacheParameterGroup>
    </CacheParameterGroups>
  </DescribeCacheParameterGroupsResult>
  <ResponseMetadata>
    <RequestId>3540cc3d-af48-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParameterGroupsResponse>
```

Viewing Parameter Values for a Cache Parameter Group

You can get a detailed listing of all parameters in a Cache Parameter Group and their values. The following procedures show you how to view the parameter values for a specific Cache Parameter Group.

AWS Management Console

To view the parameter values for a specific Cache Parameter Group

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. Click **Cache Parameter Groups** in the **Navigation** list on the left side of the window.

   The available Cache Parameter Groups appears in the **My Cache Parameter Groups** list.
3. Click on the cache parameter group that you want to see details for.

   Details about the parameters in the selected cache parameter group display in the detail pane at the bottom of the console window. You can use the navigation buttons at the top right of the detail panel to scroll through the available parameters.

You can filter the parameters shown by selecting an option from the **Viewing** drop-down list box.

CLI

To view the parameter values for a specific Cache Parameter Group

- Use the command `elasticache-describe-cache-parameters` to view the parameter values for a specific Cache Parameter Group:

```
PROMPT>elasticache-describe-cache-parameters myCacheParameterGroup --headers
```

This command should produce output similar to the following (the following example has been truncated):

```
CACHEPARAMETER  Parameter Name                  Parameter Value  Source
Data Ty
pe  Is Modifiable  Minimum Version
CACHEPARAMETER  backlog_queue_limit            1024             system
integer
false          1.4.5
CACHEPARAMETER  binding_protocol               auto             system
string
false          1.4.5
CACHEPARAMETER  cas_disabled                   0                system
boolean
true           1.4.5
CACHEPARAMETER  chunk_size                     48               system
integer
true           1.4.5
CACHEPARAMETER  chunk_size_growth_factor       1.25             system
float
true           1.4.5
CACHEPARAMETER  error_on_memory_exhausted      0                system
boolean
true           1.4.5
CACHEPARAMETER  large_memory_pages             0                system
boolean
false          1.4.5
(...sample truncated...)
```

API

To view the parameter values for a specific Cache Parameter Group

- Call `DescribeCacheParameters` with the following parameters:
  - *`CacheParameterGroupName`* = `myCacheParameterGroup`

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?CacheParameterGroupName=mycacheparametergroup
 &MaxRecords=100
 &Version=2011-07-15
 &Action=DescribeCacheParameters
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2011-07-16T01%3A17%3A51.357Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

This command should return a response similar to the following (the following example has been truncated):

```
<DescribeCacheParametersResponse xmlns="http://elmo.amazonaws.com/doc/2011-
07-15/">
  <DescribeCacheParametersResult>
    <CacheClusterClassSpecificParameters>
      <CacheNodeTypeSpecificParameter>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The maximum configurable amount of memory to use to
store i
tems, in megabytes.</Description>
        <CacheNodeTypeSpecificValues>
          <CacheNodeTypeSpecificValue>
            <Value>1000</Value>
            <CacheClusterClass>cache.c1.medium</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>6000</Value>
            <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>7100</Value>
            <CacheClusterClass>cache.m1.large</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>1300</Value>
            <CacheClusterClass>cache.m1.small</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>14600</Value>
            <CacheClusterClass>cache.m1.xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>33800</Value>
            <CacheClusterClass>cache.m2.2xlarge</CacheClusterClass>
          </CacheNodeTypeSpecificValue>
          <CacheNodeTypeSpecificValue>
            <Value>68000</Value>
            <CacheClusterClass>cache.m2.4xlarge</CacheClusterClass>
```

```
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>16600</Value>
          <CacheClusterClass>cache.m2.xlarge</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
      </CacheNodeTypeSpecificValues>
      <AllowedValues>1-100000</AllowedValues>
      <ParameterName>max_cache_memory</ParameterName>
      <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
    </CacheNodeTypeSpecificParameter>
    <CacheNodeTypeSpecificParameter>
      <DataType>integer</DataType>
      <Source>system</Source>
      <IsModifiable>false</IsModifiable>
      <Description>The number of memcached threads to use.</Description>
      <CacheNodeTypeSpecificValues>
        <CacheNodeTypeSpecificValue>
          <Value>2</Value>
          <CacheClusterClass>cache.c1.medium</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>8</Value>
          <CacheClusterClass>cache.c1.xlarge</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>2</Value>
          <CacheClusterClass>cache.m1.large</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>1</Value>
          <CacheClusterClass>cache.m1.small</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>4</Value>
          <CacheClusterClass>cache.m1.xlarge</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>4</Value>
          <CacheClusterClass>cache.m2.2xlarge</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>8</Value>
          <CacheClusterClass>cache.m2.4xlarge</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
        <CacheNodeTypeSpecificValue>
          <Value>2</Value>
          <CacheClusterClass>cache.m2.xlarge</CacheClusterClass>
        </CacheNodeTypeSpecificValue>
      </CacheNodeTypeSpecificValues>
      <AllowedValues>1-8</AllowedValues>
      <ParameterName>num_threads</ParameterName>
      <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
    </CacheNodeTypeSpecificParameter>
  </CacheClusterClassSpecificParameters>
  <Parameters>
    <Parameter>
      <ParameterValue>1024</ParameterValue>
      <DataType>integer</DataType>
```

```
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>The backlog queue limit.</Description>
        <AllowedValues>1-10000</AllowedValues>
        <ParameterName>backlog_queue_limit</ParameterName>
        <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
      </Parameter>
      <Parameter>
        <ParameterValue>auto</ParameterValue>
        <DataType>string</DataType>
        <Source>system</Source>
        <IsModifiable>false</IsModifiable>
        <Description>Binding protocol.</Description>
        <AllowedValues>auto,binary,ascii</AllowedValues>
        <ParameterName>binding_protocol</ParameterName>
        <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
      </Parameter>
      <Parameter>
        <ParameterValue>0</ParameterValue>
        <DataType>boolean</DataType>
        <Source>system</Source>
        <IsModifiable>true</IsModifiable>
       <Description>If supplied, CAS operations will be disabled, and items
 sto
red will consume 8 bytes less than with CAS enabled.</Description>
        <AllowedValues>0,1</AllowedValues>
        <ParameterName>cas_disabled</ParameterName>
        <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
      </Parameter>
      <Parameter>
        <ParameterValue>48</ParameterValue>
        <DataType>integer</DataType>
        <Source>system</Source>
        <IsModifiable>true</IsModifiable>
       <Description>The minimum amount of space to allocate for the smallest
 it
em's key + value + flags, in bytes.</Description>
        <AllowedValues>1-1024</AllowedValues>
        <ParameterName>chunk_size</ParameterName>
        <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
      </Parameter>
      <Parameter>
        <ParameterValue>1.25</ParameterValue>
        <DataType>float</DataType>
        <Source>system</Source>
        <IsModifiable>true</IsModifiable>
       <Description>The growth factor controlling the size of each successive
 m
emcached chunk - each chunk will be chunk_size_growth_factor times larger
than t
he previous chunk.</Description>
        <AllowedValues>1.01-100.00</AllowedValues>
        <ParameterName>chunk_size_growth_factor</ParameterName>
        <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
      </Parameter>
      <Parameter>
        <ParameterValue>0</ParameterValue>
        <DataType>boolean</DataType>
```

```
        <Source>system</Source>
        <IsModifiable>true</IsModifiable>
        <Description>If supplied, when there is no more memory to store
items, m
emcached will return an error rather than evicting items.</Description>
        <AllowedValues>0,1</AllowedValues>
        <ParameterName>error_on_memory_exhausted</ParameterName>
        <MinimumEngineVersion>1.4.5</MinimumEngineVersion>
      </Parameter>
    </Parameters>
  </DescribeCacheParametersResult>
  <ResponseMetadata>
    <RequestId>6d355589-af49-11e0-97f9-279771c4477e</RequestId>
  </ResponseMetadata>
</DescribeCacheParametersResponse>
```

Modifying a Cache Parameter Group

You can modify parameters in a Cache Parameter Group. These parameters are applied to Cache Clusters associated with the Cache Parameter Group when the Cache Cluster is rebooted.

In this example, you modify a parameter in a Cache Parameter Group.

AWS Management Console

To modify a Cache Parameter Group
The AWS Management Console does not support this functionality. Please refer to the command line interface example.

CLI

To modify a Cache Parameter Group

- Use the command `elasticache-modify-cache-parameter-group` to modify a Cache Parameter Group.

```
PROMPT>elasticache-modify-cache-parameter-group myCacheParameterGroup --
parameters "name=chunk_size,
value=64,name=chunk_size_growth_factor,value=1.02"
```

API

To modify a Cache Parameter Group

- Call `ModifyingCacheParameterGroup` with the following parameters:
  - *`CacheParameterGroupName`* = `mycacheparametergroup`
  - *`Parameters.member.1.ParameterName`* = `chunk_size`

- *Parameters.member.1.ParameterValue* = 64
- *Parameters.member.2.ParameterName* = chunk_size_growth_factor
- *Parameters.member.2.ParameterValue* = 1.02

Example

```
https://elasticache.us-east-1.amazonaws.com/
   ?ParameterNameValues.member.1.ParameterName=chunk_size
   &ParameterNameValues.member.1.ParameterValue=64
   &ParameterNameValues.member.2.ParameterName=chunk_size_growth_factor
   &ParameterNameValues.member.2.ParameterValue=1.02
   &CacheParameterGroupName=mycacheparametergroup
   &Version=2011-07-15
   &Action=ModifyCacheParameterGroup
   &SignatureVersion=2
   &SignatureMethod=HmacSHA256
   &Timestamp=2011-07-27T03%3A24%3A50.203Z
   &SignatureMethod=HmacSHA256
   &AWSAccessKeyId=<AWS Access Key ID>
   &Signature=<Signature>
```

This command should return a response similar to the following:

```
<ModifyCacheParameterGroupResponse xmlns="http://elasticache.amazon
aws.com/doc/2011-07-15/">
  <ModifyCacheParameterGroupResult>
    <CacheParameterGroupName>mycacheparametergroup</CacheParameterGroupName>

  </ModifyCacheParameterGroupResult>
  <ResponseMetadata>
    <RequestId>fcedeef2-b7ff-11e0-9326-b7275b9d4a6c</RequestId>
  </ResponseMetadata>
</ModifyCacheParameterGroupResponse>
```

Related Topics

- Cache Parameter Groups (p. 6)

Cache Security Groups

The following scenarios cover Cache Security Groups.

Topics

Working with Cache Security Groups

A Cache Security Group allows you to control access to your Cache Clusters. A Cache Security Group acts like a firewall controlling network access to your Cache Cluster. By default, network access is disabled for a new Cache Security Group; you must specifically authorize access to an EC2 Security Group after the Cache Security Group is created.

Creating a Cache Security Group

To create a Cache Security Group, you need to provide a name and a description.

The following procedures show you how to create a new Cache Security Group.

AWS Management Console

To create a Cache Security Group

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. Click **Cache Security Groups** in the **Navigation** list on the left side of the window.
3. Click the **Create Cache Security Group** button.
   The **Create Cache Security Group** window appears.
4. Type the name of the new Cache Security Group in the **Cache Security Group** text box.
5. Type a description for the new Cache Security Group in the **Description** text box.
6. Click the **OK** button.

CLI

To create a Cache Security Group

- Use the command `elasticache-create-cache-security-group` with the following parameters:

```
PROMPT>elasticache-create-cache-security-group mycachesecuritygroup -d "My
 new security group"
```

API

To create a Cache Security Group

- Call `CreateCacheSecurityGroup` with the following parameters:
  - *`CacheSecurityGroupName`* = `mycachesecuritygroup`
  - *`Description`* = `"My new security group"`

Example

```
https://elasticache.us-east-1.amazonaws.com /
    ?Action=CreateCacheSecurityGroup
    &CacheSecurityGroupName=mycachesecuritygroup
    &Description=My%20cache%20security%20group
    &Version=2011-07-15
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2011-07-27T02%3A43%3A10.703Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Listing Available Cache Security Groups

You can list which Cache Security Groups have been created for your AWS account.

The following procedures show you how to list the available Cache Security Groups for your AWS account.

AWS Management Console

To list all available Cache Security Groups for an AWS account

1. Launch the AWS Management Console.
   a. Go to the AWS Management Console web page.
   b. Select **Amazon ElastiCache** from the drop-down list under the **Sign in to the AWS Console** button, and then click the **Sign in to the AWS Console** button.
2. Click **Cache Security Groups** in the **Navigation** list on the left side of the window.
   The available Cache Security Groups appear in the **My Cache Security Groups** list.

CLI

To list all available Cache Security Groups for an AWS account

- Use the command `elasticache-describe-cache-security-groups` to list all available Cache Security Groups for your AWS account.

```
PROMPT>elasticache-describe-cache-security-groups
```

API

To list all available Cache Security Groups for an AWS account

- Call `DescribeCacheSecurityGroups` with no parameters.

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?Action=DescribeCacheSecurityGroups
 &MaxRecords=100
 &Version=2011-07-15
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Viewing a Cache Security Group

You can view detailed information about your Cache Security Group.

The following procedures show you how to view the properties of a Cache Security Group.

AWS Management Console

To view properties of a specific Cache Security Group

1. Launch the AWS Management Console.
   a. Go to the AWS Management Console web page.
   b. Select **Amazon ElastiCache** from the drop-down list under the **Sign in to the AWS Console** button, and then click the **Sign in to the AWS Console** button.
2. Click **Cache Security Groups** in the **Navigation** list on the left side of the window.
   The available Cache Security Groups appear in the **My Cache Security Groups** list.
3. Select a Cache Security Group from the **My Cache Security Groups** list.
   The list of authorizations defined for the Cache Security Group appears in the detail section at the bottom of the window.

CLI

To view properties of a specific Cache Security Group

- Use the `elasticache-describe-cache-security-groups` to view a Cache Security Group.

```
PROMPT>elasticache-describe-cache-security-groups mycachesecuritygroup
```

API

To view properties of a specific Cache Security Group

- Call `DescribeCacheSecurityGroups` with the following parameters:
  - *`CacheSecurityGroupName`* = `mycachesecuritygroup`

Example

```
https://elasticache.amazonaws.com/
 ?Action=DescribeCacheSecurityGroups
 &CacheParameterGroupName=mycachesecuritygroup
 &Version=2009-10-16
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2009-10-16T22%3A23%3A07.107Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Authorizing Network Access to an EC2 Security Group

If you want to access your Cache Cluster from an EC2 instance, you must grant access to the EC2 security group that the EC2 instance belongs to. The following procedures show you how to grant access to an EC2 security group.

Important

Authorizing an EC2 security group only grants access to your Cache Clusters from the EC2 instances belonging to the EC2 security group.

AWS Management Console

To grant access to an EC2 security group

1. Launch the AWS Management Console.
   a. Go to the AWS Management Console web page.
   b. Select **Amazon ElastiCache** from the drop-down list under the **Sign in to the AWS Console** button, and then click the **Sign in to the AWS Console** button.
2. Select **Cache Security Groups** from the navigation pane on the left side of the console window.
3. In the **My Cache Security Groups** list, select the check box next to the Cache Security Group that you want to grant access to.
4. In the details pane at the bottom of the window, type the name of your EC2 Security Group into the **Security Group** text box.

   

   Important

   You must enter the Group Name of the EC2 Security Group, not the Group ID. Entering the Group ID will not work.

5. Type your AWS Account ID into the **AWS Account ID** text box.

   

   Note

   You can find your AWS Account ID in the upper right corner of the Security Credentials page.

6. Click the **Add** button.

Note

It takes approximately one minute for changes to access permissions to take effect.

CLI

To grant access to an EC2 security group

- Use the command `elasticache-authorize-cache-security-group-ingress` to grant access to an EC2 security group

```
PROMPT>elasticache-authorize-cache-security-group-ingress default  --ec2-
security-group-name myec2group --ec2-security-group-owner-id 987654321021
```

The command should produce output similar to the following:

```
SECGROUP  Name     Description
SECGROUP  default  default
      EC2-SECGROUP  myec2group  987654321021  authorizing
```

API

To authorize network access to an EC2 security group

- Call `AuthorizeCacheSecurityGroupIngress` with the following parameters:
  - *`EC2SecurityGroupName`* = `myec2group`
  - *`EC2SecurityGroupOwnerId`* = `987654321021`

Example

```
https://elasticache.us-east-1.amazonaws.com/
 ?Action=AuthorizeCacheSecurityGroupIngress
 &EC2SecurityGroupOwnerId=987654321021
 &EC2SecurityGroupName=myec2group
 &Version=2009-10-16
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2009-10-22T17%3A10%3A50.274Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Related Topics

- Cache Security Groups (p. 6)

Monitoring Cache Clusters

The following scenarios cover operations for monitoring Cache Clusters.

Topics

Viewing Cache Cluster and Cache Node Metrics

Amazon ElastiCache and Amazon CloudWatch are integrated so you can gather a variety of metrics. You can monitor these metrics using Amazon CloudWatch.

Note

The following examples require the Amazon CloudWatch command line tools. For more information about Amazon CloudWatch and to download the developer tools, go to the Amazon CloudWatch product page.

The following procedures show you how to use Amazon CloudWatch to gather storage space statistics for an Cache Cluster for the past hour.

Note

The `StartTime` and `EndTime` values supplied in the examples below are for illustrative purposes. You must substitute appropriate start and end time values for your Cache Nodes.

AWS Management Console

To gather CPU utilization statistics for a Cache Cluster

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. Select the Cache Node(s) you want to view metrics for.

   a. On the **My Cache Clusters** page of the AWS management Console, click the name of one or more Cache Clusters.

      The detail page for the Cache Cluster appears.

   b. Click the **Nodes** tab at the top of the window.

   c. On the **Nodes** tab of the detail window, select the Cache Node(s) that you want to view metrics for.

      A list of available CloudWatch Metrics appears at the bottom of the console window.

   d. Click on the **CPU Utilization** metric.

      The Amazon CloudWatch console will open, displaying your selected metrics. You can use the **Statistic** and **Period** drop-down list boxes and **Time Range** tab to change the metrics being displayed.

CLI

To gather CPU utilization statistics for a Cache Cluster

- Use the Amazon CloudWatch command **mon-get-stats** with the following parameters (note that the start end end times are shown as examples only; you will need to substitute your own appropriate start and end times):

```
PROMPT> mon-get-stats CPUUtilization --dimensions="CacheClusterId=my
cachecluster,CacheNodeId=0002" --statistics=Average
--namespace="AWS/ElastiCache" --start-time 2011-12-05T00:00:00 --end-time
2011-12-06T00:00:00 --period=60
```

API

To gather CPU utilization statistics for a Cache Cluster

- Call the Amazon CloudWatch API `GetMetricStatistics` with the following parameters (note that the start end end times are shown as examples only; you will need to substitute your own appropriate start and end times):
  - *`Statistics.member.1`* = `Average`
  - *`Namespace`* = `AWS/ElastiCache`
  - *`StartTime`* = `2011-12-05T00:00:00`
  - *`EndTime`* = `2011-12-06T00:00:00`
  - *`Period`* = `60`
  - *`MeasureName`* = `CPUUtilization`
  - *`Dimensions`* = `CacheClusterId=mycachecluster,CacheNodeId=0002`

  Example

```
http://monitoring.amazonaws.com/
 ?SignatureVersion=2
 &Action=GetMetricStatistics
 &Version=2011-07-15
 &StartTime=2011-07-16T00:00:00
 &EndTime=2011-07-16T00:02:00
 &Period=60
 &Statistics.member.1=Average
 &Dimensions.member.1="CacheClusterId=mycachecluster"
 &Dimensions.member.2="CacheNodeId=0002"
 &Namespace=AWS/ElastiCache
 &MeasureName=CPUUtilization
 &Timestamp=2011-12-07T17%3A48%3A21.746Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Related Topics

- Creating and Modifying Cache Clusters (p. 32)

  CloudWatch Metrics with Amazon ElastiCache (p. 13)

Working with Amazon ElastiCache Events

The following scenarios cover operations for viewing Cache Cluster events.

Topics

Viewing Amazon ElastiCache Events

Amazon ElastiCache logs events that relate to your Cache Instances, Cache Security Groups, and Cache Parameter Groups. This information includes the date and time of the event, the source name and source type of the event, and a description of the event. You can easily retrieve events from the log using the `elasticache-describe-events command` or the `DescribeEvents` API.

The following procedures show you how to view all Amazon ElastiCache events for the past 24 hours (specified in seconds).

AWS Management Console

To view all Amazon ElastiCache instance events for the past 24 hours

1. Sign in to the AWS Management Console and open the Amazon ElastiCache console at https://console.aws.amazon.com/elasticache/.
2. Click **Cache Events** in the **Navigation** list on the left side of the window.
   The available events appear in the **My Cache Events** list.

Note

You can use the **Viewing** drop-down list box to filter the events by type.

CLI

To view all Amazon ElastiCache instance events for the past 24 hours

- Use the command `elasticache-describe-events` with the following parameters to view all Amazon ElastiCache events for the past 24 hours.

```
PROMPT>elasticache-describe-events --duration 1440
```

API

To view all Amazon ElastiCache instance events for the past 24 hours

- Call `DescribeEvents` with the following parameters:
  - *`Duration`* = `1440`

Example

```
https://elasticache.amazonaws.com/
 ?Action=DescribeEvents
 &Duration=1440
 &MaxRecords=100
 &Version=2011-07-16
 &SignatureVersion=2
 &SignatureMethod=HmacSHA256
 &Timestamp=2009-10-22T20%3A00%3A44.420Z
 &AWSAccessKeyId=<AWS Access Key ID>
 &Signature=<Signature>
```

Related Topics

- Using Amazon Simple Notification Service (SNS) with Amazon ElastiCache (p. 15)

Document History

This What's New is associated with the 2011-07-15 version of Amazon ElastiCache. This guide was last updated on 05 March 2012.

The following table describes the important changes since the last release of the *Amazon ElastiCache User Guide*.

Change	Description	Release Date
New Guide	This is the first release of *Amazon ElastiCache User Guide*.	22 August 2011

Amazon ElastiCache Resources

The following table lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon ElastiCache Getting Started Guide	The Getting Started Guide provides a quick tutorial on using the Amazon ElastiCache console based on short, simple examples.
Amazon ElastiCache API Reference	The API Reference contains a comprehensive description of all Amazon ElastiCache Query APIs and data types.
Amazon ElastiCache Command Line Interface Reference	The Command Line Tools Reference contains a comprehensive description of all the command line tools and their options.
Amazon ElastiCache Technical FAQ	The FAQ covers the top questions developers have asked about this product.
Release notes	The release notes give a high-level overview of the current release. They specifically note any new features, corrections, and known issues.
AWS Developer Resource Center	A central starting point to find documentation, code samples, release notes, and other information to help you build innovative applications with AWS.
AWS Management Console	The AWS Management Console allows you to perform most of the functions of Amazon ElastiCache without programming.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon Web Services.
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support.
Amazon ElastiCache product information	The primary web page for information about Amazon ElastiCache.

Resource	Description
Contact Us	A central contact point for inquiries concerning AWS billing, account, events, abuse etc.
Conditions of Use	Detailed information about the copyright and trademark usage at Amazon.com and other topics.

Glossary

Glossary

Availability Zone	Amazon ElastiCache uses the same locations as Amazon EC2. These locations are composed of Regions and Availability Zones.
Cache Engine Version	The version of memcached running on the cache node.
Cache Cluster	A logical cache distributed over multiple Cache Nodes. A Cache Cluster can be set up with a specific number of *Cache Nodes*.
Cache Node	A fixed-size chunk of secure, network-attached RAM. Each Cache Node runs an instance of the Memcached service, and has its own DNS name and port. Multiple types of Cache Nodes are supported, each with varying amounts of associated memory.
Cache Node Type	EC2 Instance Type used to run the Cache Node.
Cache Cluster Identifier	Customer supplied identifier for the Cache Cluster that must be unique for that customer in an AWS region.
Cache Parameter Group	A container for cache engine parameter values that can be applied to one or more Cache Clusters.
Cache Security Group	A Cache Security Group is a group maintained by Amazon ElastiCache that combines ingress authorizations to Cache Nodes for hosts belonging to EC2 Security Groups specified through the console or the API or command line tools.
EC2 Compute Unit	An AWS standard for compute CPU and memory. One EC2 Compute Unit (ECU) provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

Endpoint	DNS name of a Cache Node.
Endpoint Port	Port number used by a Cache Node. See Endpoint.
Marker	See pagination
Pagination	APIs that return a potentially large list of records allow a subset to be viewed by using a *MaxRecords* and *Marker* value. The *MaxRecords* value corresponds to the maximum number of records to return, in the event *MaxRecords* is not specified a default is used. The *Marker* identifies the last record returned in the set of records (if the set is larger than *MaxRecords*).
Quota	Amazon ElastiCache has quotas for: • The number of Cache Clusters for each AWS account • The number of Cache Nodes per Cache Cluster • The total number of Cache Nodes per AWS account across all Cache Clusters created by that AWS account For more information on Amazon ElastiCache quotas, go to the Amazon ElastiCache product page.