# Table of Contents

Welcome ........................................................................................................................................... 1  
AWS Elemental MediaStore ............................................................................................................. 1  
AWS Elemental MediaStore Data Plane ............................................................................................. 1  
Actions ............................................................................................................................................... 2  
  AWS Elemental MediaStore ......................................................................................................... 2  
    CreateContainer ......................................................... 4  
    DeleteContainer .......................................................... 7  
    DeleteContainerPolicy .................................................. 9  
    DeleteCorsPolicy .......................................................... 11  
    DeleteLifecyclePolicy ..................................................... 13  
    DescribeContainer ..................................................... 15  
    GetContainerPolicy ..................................................... 17  
    GetCorsPolicy ............................................................. 19  
    GetLifecyclePolicy ....................................................... 21  
    ListContainers ............................................................ 23  
    PutContainerPolicy ..................................................... 26  
    PutCorsPolicy .............................................................. 28  
    PutLifecyclePolicy ....................................................... 30  
    StartAccessLogging ...................................................... 32  
    StopAccessLogging ....................................................... 34  
AWS Elemental MediaStore Data Plane ............................................................................................. 35  
  DeleteObject ................................................................. 36  
  DescribeObject ............................................................... 38  
  GetObject ................................................................. 41  
  ListItems ................................................................. 45  
  PutObject ................................................................. 48  
Data Types ........................................................................................................................................... 52  
  AWS Elemental MediaStore ..................................................... 52  
    Container ................................................................. 53  
    CorsRule ................................................................. 55  
AWS Elemental MediaStore Data Plane ............................................................................................. 56  
  Item ............................................................................. 57  
Common Parameters .......................................................................................................................... 59  
Common Errors ............................................................................................................................... 61
Welcome

AWS Elemental MediaStore

An AWS Elemental MediaStore container is a namespace that holds folders and objects. You use a container endpoint to create, read, and delete objects.

AWS Elemental MediaStore Data Plane

An AWS Elemental MediaStore asset is an object, similar to an object in the Amazon S3 service. Objects are the fundamental entities that are stored in AWS Elemental MediaStore.
Actions

The following actions are supported by AWS Elemental MediaStore:

• CreateContainer (p. 4)
• DeleteContainer (p. 7)
• DeleteContainerPolicy (p. 9)
• DeleteCorsPolicy (p. 11)
• DeleteLifecyclePolicy (p. 13)
• DescribeContainer (p. 15)
• GetContainerPolicy (p. 17)
• GetCorsPolicy (p. 19)
• GetLifecyclePolicy (p. 21)
• ListContainers (p. 23)
• PutContainerPolicy (p. 26)
• PutCorsPolicy (p. 28)
• PutLifecyclePolicy (p. 30)
• StartAccessLogging (p. 32)
• StopAccessLogging (p. 34)

The following actions are supported by AWS Elemental MediaStore Data Plane:

• DeleteObject (p. 36)
• DescribeObject (p. 38)
• GetObject (p. 41)
• ListItems (p. 45)
• PutObject (p. 48)

AWS Elemental MediaStore

The following actions are supported by AWS Elemental MediaStore:

• CreateContainer (p. 4)
• DeleteContainer (p. 7)
• DeleteContainerPolicy (p. 9)
• DeleteCorsPolicy (p. 11)
• DeleteLifecyclePolicy (p. 13)
• DescribeContainer (p. 15)
• GetContainerPolicy (p. 17)
• GetCorsPolicy (p. 19)
• GetLifecyclePolicy (p. 21)
• ListContainers (p. 23)
• PutContainerPolicy (p. 26)
• PutCorsPolicy (p. 28)
• **PutLifecyclePolicy** (p. 30)
• **StartAccessLogging** (p. 32)
• **StopAccessLogging** (p. 34)
CreateContainer
Service: AWS Elemental MediaStore

Creates a storage container to hold objects. A container is similar to a bucket in the Amazon S3 service.

Request Syntax

```
{
   "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 4)**

The name for the container. The name must be from 1 to 255 characters. Container names must be unique to your AWS account within a specific region. As an example, you could create a container named movies in every region, as long as you don’t have an existing container with that name.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w\-\]+

Required: Yes

Response Syntax

```
{
   "Container": {
      "AccessLoggingEnabled": boolean,
      "ARN": "string",
      "CreationTime": number,
      "Endpoint": "string",
      "Name": "string",
      "Status": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**Container (p. 4)**

ContainerARN: The Amazon Resource Name (ARN) of the newly created container. The ARN has the following format: arn:aws:<region>:<account that owns this container>:container/<name of container>. For example: arn:aws:mediastore:us-west-2:111122223333:container/movies
ContainerName: The container name as specified in the request.

CreationTime: Unix time stamp.

Status: The status of container creation or deletion. The status is one of the following: CREATING, ACTIVE, or DELETING. While the service is creating the container, the status is CREATING. When an endpoint is available, the status changes to ACTIVE.

The return value does not include the container's endpoint. To make downstream requests, you must obtain this value by using DescribeContainer (p. 15) or ListContainers (p. 23).

Type: Container (p. 53) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

ContainerInUseException

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

LimitExceededError

A service limit has been exceeded.

HTTP Status Code: 400

Examples

Example Request

The following example shows how to create a container with the name movies:

```
POST / HTTP/1.1
Host: mediastore.us-west-2.amazonaws.com
x-amz-date: 20170323T120000Z
x-amz-target:ElementalMediaStoreService_20160711.CreateContainer
content-type:application/x-amz-json-1.1
Content-Length:30
Authorization: AWS4-HMAC-SHA256
Credential=AKIAIOSFODNN7EXAMPLE/20141123/us-west-2/mediastore/aws4_request,
SignedHeaders=host;x-amz-date;x-amz-mediastore-version,
Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
{
  "ContainerName": "movies",
}
```

Example Response

```
HTTP/1.1
```

API Version 2017-09-01
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
**DeleteContainer**

Service: AWS Elemental MediaStore

Deletes the specified container. Before you make a DeleteContainer request, delete any objects in the container or in any folders in the container. You can delete only empty containers.

**Request Syntax**

```json
{
  "ContainerName": "string"
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 7)**

The name of the container to delete.

- **Type:** String
- **Length Constraints:** Minimum length of 1. Maximum length of 255.
- **Pattern:** `[\w-]`
- **Required:** Yes

**Response Elements**

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerInUseException**

The container that you specified in the request already exists or is being updated.

- **HTTP Status Code:** 400

**ContainerNotFoundException**

The container that you specified in the request does not exist.

- **HTTP Status Code:** 400

**InternalServerError**

The service is temporarily unavailable.

- **HTTP Status Code:** 500
Example

Example Request

The following command has no return value:

```plaintext
POST / HTTP/1.1
content-type:application/x-amz-json-1.1
host:mediastore.us-west-2.amazonaws.com
x-amz-date:20170620T230417Z
x-amz-target:ElementalMediaStoreService_20160711.DeleteContainer
{
  "ContainerName":"movies"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteContainerPolicy
Service: AWS Elemental MediaStore

Deletes the access policy that is associated with the specified container.

Request Syntax

```
{
   "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 9)**

The name of the container that holds the policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w\-\]+

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerInUseException**

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

**ContainerNotFoundException**

The container that you specified in the request does not exist.

HTTP Status Code: 400

**InternalServerError**

The service is temporarily unavailable.

HTTP Status Code: 500

**PolicyNotFoundException**

The policy that you specified in the request does not exist.
HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteCorsPolicy
Service: AWS Elemental MediaStore

Deletes the cross-origin resource sharing (CORS) configuration information that is set for the container.

To use this operation, you must have permission to perform the MediaStore:DeleteCorsPolicy action. The container owner has this permission by default and can grant this permission to others.

Request Syntax

```
{
   "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

containerName (p. 11)

The name of the container to remove the policy from.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w-]+

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

ContainerInUseException

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

ContainerNotFoundException

The container that you specified in the request does not exist.

HTTP Status Code: 400

CorsPolicyNotFoundException

The CORS policy that you specified in the request does not exist.

HTTP Status Code: 400
InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DeleteLifecyclePolicy
Service: AWS Elemental MediaStore

Removes an object lifecycle policy from a container. It takes up to 20 minutes for the change to take effect.

Request Syntax

```
{
   "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 13)**

The name of the container that holds the object lifecycle policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w-]+

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerInUseException**

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

**ContainerNotFoundException**

The container that you specified in the request does not exist.

HTTP Status Code: 400

**InternalServerError**

The service is temporarily unavailable.

HTTP Status Code: 500

**PolicyNotFoundException**

The policy that you specified in the request does not exist.
HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeContainer
Service: AWS Elemental MediaStore

Retrieves the properties of the requested container. This request is commonly used to retrieve the endpoint of a container. An endpoint is a value assigned by the service when a new container is created. A container's endpoint does not change after it has been assigned. The DescribeContainer request returns a single Container object based on ContainerName. To return all Container objects that are associated with a specified AWS account, use ListContainers (p. 23).

Request Syntax

```json
{
  "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

ContainerName (p. 15)

The name of the container to query.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w-]+

Required: No

Response Syntax

```json
{
  "Container": {
    "AccessLoggingEnabled": boolean,
    "ARN": "string",
    "CreationTime": number,
    "Endpoint": "string",
    "Name": "string",
    "Status": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Container (p. 15)

The name of the queried container.
Type: Container (p. 53) object

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerNotFoundException**

The container that you specified in the request does not exist.

HTTP Status Code: 400

**InternalServer Error**

The service is temporarily unavailable.

HTTP Status Code: 500

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetContainerPolicy
Service: AWS Elemental MediaStore

Retrieves the access policy for the specified container. For information about the data that is included in an access policy, see the AWS Identity and Access Management User Guide.

Request Syntax

```
{
    "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 17)**

The name of the container.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: [\w-]+

Required: Yes

Response Syntax

```
{
    "Policy": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**Policy (p. 17)**

The contents of the access policy.

Type: String


Pattern: [\u0009\u000A\u000D\u0020-\u00FF]+

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).
**ContainerInUseException**

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

**ContainerNotFoundException**

The container that you specified in the request does not exist.

HTTP Status Code: 400

**InternalServerError**

The service is temporarily unavailable.

HTTP Status Code: 500

**PolicyNotFoundException**

The policy that you specified in the request does not exist.

HTTP Status Code: 400

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetCorsPolicy
Service: AWS Elemental MediaStore

Returns the cross-origin resource sharing (CORS) configuration information that is set for the container.

To use this operation, you must have permission to perform the `MediaStore:GetCorsPolicy` action. By default, the container owner has this permission and can grant it to others.

Request Syntax

```
{
    "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 19)**

The name of the container that the policy is assigned to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w-]+

Required: Yes

Response Syntax

```
{
    "CorsPolicy": [
        {
            "AllowedHeaders": [ "string" ],
            "AllowedMethods": [ "string" ],
            "AllowedOrigins": [ "string" ],
            "ExposeHeaders": [ "string" ],
            "MaxAgeSeconds": number
        }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**CorsPolicy (p. 19)**

The CORS policy assigned to the container.
Type: Array of CorsRule (p. 55) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

ContainerInUseException

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

ContainerNotFoundException

The container that you specified in the request does not exist.

HTTP Status Code: 400

CorsPolicyNotFoundException

The CORS policy that you specified in the request does not exist.

HTTP Status Code: 400

InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetLifecyclePolicy

Service: AWS Elemental MediaStore

Retrieves the object lifecycle policy that is assigned to a container.

Request Syntax

```json
{
  "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 21)**

The name of the container that the object lifecycle policy is assigned to.

- Type: String
- Pattern: \[\w-]+ 
- Required: Yes

Response Syntax

```json
{
  "LifecyclePolicy": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**LifecyclePolicy (p. 21)**

The object lifecycle policy that is assigned to the container.

- Type: String
- Length Constraints: Minimum length of 0. Maximum length of 8192.
- Pattern: \[\u0009\u000A\u000D\u0020-\u00FF]+ 

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).
ContainerInUseException

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

ContainerNotFoundException

The container that you specified in the request does not exist.

HTTP Status Code: 400

InternalServerException

The service is temporarily unavailable.

HTTP Status Code: 500

PolicyNotFoundException

The policy that you specified in the request does not exist.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
ListContainers
Service: AWS Elemental MediaStore

Lists the properties of all containers in AWS Elemental MediaStore.

You can query to receive all the containers in one response. Or you can include the `MaxResults` parameter to receive a limited number of containers in each response. In this case, the response includes a token. To get the next set of containers, send the command again, this time with the `NextToken` parameter (with the returned token as its value). The next set of responses appears, with a token if there are still more containers to receive.

See also DescribeContainer (p. 15), which gets the properties of one container.

Request Syntax

```json
{
    "MaxResults": number,
    "NextToken": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**MaxResults (p. 23)**

Enter the maximum number of containers in the response. Use from 1 to 255 characters.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

**NextToken (p. 23)**

Only if you used `MaxResults` in the first command, enter the token (which was included in the previous response) to obtain the next set of containers. This token is included in a response only if there actually are more containers to list.

Type: String


Pattern: `[^\0-9A-Za-z/=+/]+`

Required: No

Response Syntax

```json
{
    "Containers": [
        {
            "AccessLoggingEnabled": boolean,
            ...
        }
    ]
}
```
"ARN": "string",
"CreationTime": number,
"Endpoint": "string",
"Name": "string",
"Status": "string"
};
}
"NextToken": "string"
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Containers (p. 23)

The names of the containers.

Type: Array of Container (p. 53) objects

NextToken (p. 23)

NextToken is the token to use in the next call to ListContainers. This token is returned only if you included the MaxResults tag in the original command, and only if there are still containers to return.

Type: String


Pattern: [0-9A-Za-z=/+]+

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutContainerPolicy
Service: AWS Elemental MediaStore

Creates an access policy for the specified container to restrict the users and clients that can access it. For information about the data that is included in an access policy, see the AWS Identity and Access Management User Guide.

For this release of the REST API, you can create only one policy for a container. If you enter PutContainerPolicy twice, the second command modifies the existing policy.

Request Syntax

```
{
  "ContainerName": "string",
  "Policy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 26)**

The name of the container.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w-]+

Required: Yes

**Policy (p. 26)**

The contents of the policy, which includes the following:

- One `Version` tag
- One `Statement` tag that contains the standard tags for the policy.

Type: String


Pattern: \[\u0009\u000A\u000D\u0020-\u00FF]+

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).
ContainerInUseException

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

ContainerNotFoundException

The container that you specified in the request does not exist.

HTTP Status Code: 400

InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutCorsPolicy

Service: AWS Elemental MediaStore

Sets the cross-origin resource sharing (CORS) configuration on a container so that the container can service cross-origin requests. For example, you might want to enable a request whose origin is http://www.example.com to access your AWS Elemental MediaStore container at my.example.container.com by using the browser's XMLHttpRequest capability.

To enable CORS on a container, you attach a CORS policy to the container. In the CORS policy, you configure rules that identify origins and the HTTP methods that can be executed on your container. The policy can contain up to 398,000 characters. You can add up to 100 rules to a CORS policy. If more than one rule applies, the service uses the first applicable rule listed.

To learn more about CORS, see Cross-Origin Resource Sharing (CORS) in AWS Elemental MediaStore.

Request Syntax

```json
{
  "ContainerName": "string",
  "CorsPolicy": [
    {
      "AllowedHeaders": [ "string" ],
      "AllowedMethods": [ "string" ],
      "AllowedOrigins": [ "string" ],
      "ExposeHeaders": [ "string" ],
      "MaxAgeSeconds": number
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 28)**

The name of the container that you want to assign the CORS policy to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: `\w-`+

Required: Yes

**CorsPolicy (p. 28)**

The CORS policy to apply to the container.

Type: Array of CorsRule (p. 55) objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: Yes
Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

ContainerInUseException

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

ContainerNotFoundException

The container that you specified in the request does not exist.

HTTP Status Code: 400

InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
PutLifecyclePolicy
Service: AWS Elemental MediaStore

Writes an object lifecycle policy to a container. If the container already has an object lifecycle policy, the service replaces the existing policy with the new policy. It takes up to 20 minutes for the change to take effect.

For information about how to construct an object lifecycle policy, see Components of an Object Lifecycle Policy.

Request Syntax

```
{
    "ContainerName": "string",
    "LifecyclePolicy": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 30)**

The name of the container that you want to assign the object lifecycle policy to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Pattern: \[\w-]+

Required: Yes

**LifecyclePolicy (p. 30)**

The object lifecycle policy to apply to the container.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 8192.

Pattern: \[\u0009\u000A\u000D\u0020-\u00FF\]+

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerInUseException**

The container that you specified in the request already exists or is being updated.
HTTP Status Code: 400

ContainerNotFoundException

The container that you specified in the request does not exist.

HTTP Status Code: 400

InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
StartAccessLogging
Service: AWS Elemental MediaStore

Starts access logging on the specified container. When you enable access logging on a container, MediaStore delivers access logs for objects stored in that container to Amazon CloudWatch Logs.

Request Syntax

```json
{
    "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

ContainerName (p. 32)

- The name of the container that you want to start access logging on.
- Type: String
- Pattern: \[\w\-]+
- Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

ContainerInUseException

- The container that you specified in the request already exists or is being updated.
- HTTP Status Code: 400

ContainerNotFoundException

- The container that you specified in the request does not exist.
- HTTP Status Code: 400

InternalServerError

- The service is temporarily unavailable.
- HTTP Status Code: 500
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
StopAccessLogging
Service: AWS Elemental MediaStore

Stops access logging on the specified container. When you stop access logging on a container, MediaStore stops sending access logs to Amazon CloudWatch Logs. These access logs are not saved and are not retrievable.

Request Syntax

```
{
    "ContainerName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 59).

The request accepts the following data in JSON format.

**ContainerName (p. 34)**

The name of the container that you want to stop access logging on.

- Type: String
- Pattern: `[\w-]+`
- Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerInUseException**

The container that you specified in the request already exists or is being updated.

HTTP Status Code: 400

**ContainerNotFoundException**

The container that you specified in the request does not exist.

HTTP Status Code: 400

**InternalServerException**

The service is temporarily unavailable.

HTTP Status Code: 500
See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2

AWS Elemental MediaStore Data Plane

The following actions are supported by AWS Elemental MediaStore Data Plane:

- DeleteObject (p. 36)
- DescribeObject (p. 38)
- GetObject (p. 41)
- ListItems (p. 45)
- PutObject (p. 48)
DeleteObject
Service: AWS Elemental MediaStore Data Plane
Deletes an object at the specified path.

Request Syntax
```
DELETE /Path+ HTTP/1.1
```

URI Request Parameters
The request requires the following URI parameters.

Path (p. 36)
The path (including the file name) where the object is stored in the container. Format: <folder name>/<folder name>/<file name>


Pattern: (?:[A-Za-z0-9_.\-~]+/){0,10}[A-Za-z0-9_.\-~]+(0,10)[A-Za-z0-9_.\-~]+

Request Body
The request does not have a request body.

Response Syntax
```
HTTP/1.1 200
```

Response Elements
If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors
For information about the errors that are common to all actions, see Common Errors (p. 61).

ContainerNotFoundException
The specified container was not found for the specified account.

HTTP Status Code: 404

InternalServerError
The service is temporarily unavailable.

HTTP Status Code: 500

ObjectNotFoundException
Could not perform an operation on an object that does not exist.

HTTP Status Code: 404
Example

The following request deletes the file `mlaw.avi` that is in the folder `premium/canada`:

```
DELETE premium/canada/mlaw.avi
Host: aaabbbccccdddee.files.mediastore-us-west-2.com
x-amz-Date: 20170323T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/us-west-2/mediastore/aws4_request,SignedHeaders=host;x-amz-date;x-amz-mediastore-version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dc54091b12966f2a2
Content-Length: 0
x-amz-mediastore-version: 2016-07-11
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
DescribeObject
Service: AWS Elemental MediaStore Data Plane

Gets the headers for an object at the specified path.

Request Syntax

```plaintext
HEAD /Path+ HTTP/1.1
```

URI Request Parameters

The request requires the following URI parameters.

**Path (p. 38)**

The path (including the file name) where the object is stored in the container. Format: `<folder name>/<folder name>/<file name>`


Pattern: `(\?:[A-Za-z0-9\._\-~]+/){0,10}[A-Za-z0-9\._\-~]+`

Request Body

The request does not have a request body.

Response Syntax

```plaintext
HTTP/1.1 200
ETag: ETag
Content-Type: ContentType
Content-Length: ContentLength
Cache-Control: CacheControl
Last-Modified: LastModified
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The response returns the following HTTP headers.

**CacheControl (p. 38)**

An optional CacheControl header that allows the caller to control the object’s cache behavior. Headers can be passed in as specified in the HTTP at `https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9`.

Headers with a custom user-defined value are also accepted.

**ContentLength (p. 38)**

The length of the object in bytes.

Valid Range: Minimum value of 0.
**ContentType (p. 38)**

The content type of the object.

Pattern: `^[\w\-\_/\.&+]{1,255}$`

**ETag (p. 38)**

The ETag that represents a unique instance of the object.

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `[0-9A-Fa-f]+`

**LastModified (p. 38)**

The date and time that the object was last modified.

---

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerNotFoundException**

The specified container was not found for the specified account.

HTTP Status Code: 404

**InternalServerError**

The service is temporarily unavailable.

HTTP Status Code: 500

**ObjectNotFoundException**

Could not perform an operation on an object that does not exist.

HTTP Status Code: 404

---

**Example**

The following request gets the headers for the file mlaw.avi from the folder premium/canada in the container that is identified by the endpoint that is specified in the Host: header.

```
HEAD premium/canada/mlaw.avi
Host: aabbbccccdddee.files.mediastore-us-west-2.com
x-amz-Date: 20170323T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/us-west-2/mediastore/aws4_request, SignedHeaders=host;x-amz-date;x-amz-mediastore-version, Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: 0
x-amz-mediastore-version: 2016-07-11
```

---

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
GetObject

Service: AWS Elemental MediaStore Data Plane

Downloads the object at the specified path. If the object's upload availability is set to streaming, AWS Elemental MediaStore downloads the object even if it's still uploading the object.

Request Syntax

GET /Path HTTP/1.1
Range: Range

URI Request Parameters

The request requires the following URI parameters.

Path (p. 41)

The path (including the file name) where the object is stored in the container. Format: <folder name>/<folder name>/<file name>

For example, to upload the file mlaw.avi to the folder path premium\canada in the container movies, enter the path premium/canada/mlaw.avi.

Do not include the container name in this path.

If the path includes any folders that don't exist yet, the service creates them. For example, suppose you have an existing premium/usa subfolder. If you specify premium/canada, the service creates a canada subfolder in the premium folder. You then have two subfolders, usa and canada, in the premium folder.

There is no correlation between the path to the source and the path (folders) in the container in AWS Elemental MediaStore.

For more information about folders and how they exist in a container, see the AWS Elemental MediaStore User Guide.

The file name is the name that is assigned to the file that you upload. The file can have the same name inside and outside of AWS Elemental MediaStore, or it can have the same name. The file name can include or omit an extension.


Pattern: (?:[A-Za-z0-9_\-\~\]+/){0,10}[A-Za-z0-9_\-\~\]+

Range (p. 41)

The range bytes of an object to retrieve. For more information about the Range header, see http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.35. AWS Elemental MediaStore ignores this header for partially uploaded objects that have streaming upload availability.

Pattern: ^bytes=(?:\d+-\d*|\d*\-\d+)$

Request Body

The request does not have a request body.
Response Syntax

<table>
<thead>
<tr>
<th>HTTP/1.1</th>
<th>StatusCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache-Control:</td>
<td>CacheControl</td>
</tr>
<tr>
<td>Content-Range:</td>
<td>ContentRange</td>
</tr>
<tr>
<td>Content-Length:</td>
<td>ContentLength</td>
</tr>
<tr>
<td>Content-Type:</td>
<td>ContentType</td>
</tr>
<tr>
<td>ETag:</td>
<td>ETag</td>
</tr>
<tr>
<td>Last-Modified:</td>
<td>LastModified</td>
</tr>
</tbody>
</table>

Body

Response Elements

If the action is successful, the service sends back the following HTTP response.

**StatusCode (p. 42)**

The HTML status code of the request. Status codes ranging from 200 to 299 indicate success. All other status codes indicate the type of error that occurred.

The response returns the following HTTP headers.

**CacheControl (p. 42)**

An optional CacheControl header that allows the caller to control the object's cache behavior. Headers can be passed in as specified in the HTTP spec at https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9.

Headers with a custom user-defined value are also accepted.

**ContentLength (p. 42)**

The length of the object in bytes.

Valid Range: Minimum value of 0.

**ContentRange (p. 42)**

The range of bytes to retrieve.

Pattern: ^bytes=\d+-\d+/\d+$

**ContentType (p. 42)**

The content type of the object.

Pattern: ^[\w\-\./\+]{1,255}$

**ETag (p. 42)**

The ETag that represents a unique instance of the object.

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: [0-9A-Fa-f]+

**LastModified (p. 42)**

The date and time that the object was last modified.

The response returns the following as the HTTP body.
Body (p. 42)

The bytes of the object.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

ContainerNotFoundException

The specified container was not found for the specified account.

HTTP Status Code: 404

InternalServerError

The service is temporarily unavailable.

HTTP Status Code: 500

ObjectNotFoundException

Could not perform an operation on an object that does not exist.

HTTP Status Code: 404

RequestedRangeNotSatisfiableException

The requested content range is not valid.

HTTP Status Code: 416

Example

The following request downloads the file mlaw.avi from the folder premium/canada in the container that is identified by the endpoint that is specified in the Host: header.

```
GET premium/canada/mlaw.avi
Host: aaabbbcccddee.files.mediastore-us-west-2.com
x-amz-date: 20170323T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/us-west-2/mediastore/aws4_request,SignedHeaders=host;x-amz-date;x-amz-mediastore-version,Signature=9257c16da6b25a715ce900a5b45b03da0447acf430195dcb540091b12966f2a2
Content-Length: 0
x-amz-mediastore-version: 2016-07-11
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
ListItems
Service: AWS Elemental MediaStore Data Plane

Provides a list of metadata entries about folders and objects in the specified folder.

Request Syntax

GET /?MaxResults=MaxResults&NextToken=NextToken&Path=Path HTTP/1.1

URI Request Parameters

The request requires the following URI parameters.

MaxResults (p. 45)

The maximum number of results to return per API request. For example, you submit a ListItems request with MaxResults set at 500. Although 2,000 items match your request, the service returns no more than the first 500 items. (The service also returns a NextToken value that you can use to fetch the next batch of results.) The service might return fewer results than the MaxResults value.

If MaxResults is not included in the request, the service defaults to pagination with a maximum of 1,000 results per page.

Valid Range: Minimum value of 1. Maximum value of 1000.

NextToken (p. 45)

The token that identifies which batch of results that you want to see. For example, you submit a ListItems request with MaxResults set at 500. The service returns the first batch of results (up to 500) and a NextToken value. To see the next batch of results, you can submit the ListItems request a second time and specify the NextToken value.

Tokens expire after 15 minutes.

Path (p. 45)

The path in the container from which to retrieve items. Format: <folder name>/<folder name>/<file name>

Length Constraints: Minimum length of 0. Maximum length of 900.

Pattern: /?(?:[A-Za-z0-9_\-\.\~\+]?)(0,10)(?:[A-Za-z0-9_\-\.\~\+]?)/?

Request Body

The request does not have a request body.

Response Syntax

HTTP/1.1 200
Content-type: application/json

{ 
  "Items": [ 
    { 
      "ContentLength": number,
      "ContentType": "string",
      "Name": "string",
      "Path": "string",
      "Version": "string"
    }
  ]
}
"ETag": "string",
"LastModified": number,
"Name": "string",
"Type": "string"
}

"NextToken": "string"
}

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

**Items (p. 45)**

The metadata entries for the folders and objects at the requested path.

Type: Array of Item (p. 57) objects

**NextToken (p. 45)**

The token that can be used in a request to view the next set of results. For example, you submit a ListItems request that matches 2,000 items with MaxResults set at 500. The service returns the first batch of results (up to 500) and a NextToken value that can be used to fetch the next batch of results.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerNotFoundException**

The specified container was not found for the specified account.

HTTP Status Code: 404

**InternalServerError**

The service is temporarily unavailable.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
PutObject
Service: AWS Elemental MediaStore Data Plane

Uploads an object to the specified path. Object sizes are limited to 25 MB for standard upload availability and 10 MB for streaming upload availability.

Request Syntax

```
PUT /Path+ HTTP/1.1
Content-Type: ContentType
Cache-Control: CacheControl
x-amz-storage-class: StorageClass
x-amz-upload-availability: UploadAvailability

Body
```

URI Request Parameters

The request requires the following URI parameters.

**CacheControl (p. 48)**

An optional CacheControl header that allows the caller to control the object's cache behavior. Headers can be passed in as specified in the HTTP at https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9.

Headers with a custom user-defined value are also accepted.

**ContentType (p. 48)**

The content type of the object.

Pattern: ^\w\-\//\./.+\{1,255}$

**Path (p. 48)**

The path (including the file name) where the object is stored in the container. Format: <folder name>/<folder name>/<file name>

For example, to upload the file mlaw.avi to the folder path premium\canada in the container movies, enter the path premium/canada/mlaw.avi.

Do not include the container name in this path.

If the path includes any folders that don't exist yet, the service creates them. For example, suppose you have an existing premium\usa subfolder. If you specify premium/canada, the service creates a canada subfolder in the premium folder. You then have two subfolders, usa and canada, in the premium folder.

There is no correlation between the path to the source and the path (folders) in the container in AWS Elemental MediaStore.

For more information about folders and how they exist in a container, see the AWS Elemental MediaStore User Guide.

The file name is the name that is assigned to the file that you upload. The file can have the same name inside and outside of AWS Elemental MediaStore, or it can have the same name. The file name can include or omit an extension.

PutObject

Pattern: (?:[A-Za-z0-9_\-.\-~]+/){0,10}[A-Za-z0-9_\-.\-~]+ StorageClass (p. 48)

Indicates the storage class of a Put request. Defaults to high-performance temporal storage class, and objects are persisted into durable storage shortly after being received.


Valid Values: STANDARD | TEMPORAL

UploadAvailability (p. 48)

Indicates the availability of an object while it is still uploading. If the value is set to streaming, the object is available for downloading after some initial buffering but before the object is uploaded completely. If the value is set to standard, the object is available for downloading only when it is uploaded completely. The default value for this header is standard.

To use this header, you must also set the HTTP Transfer-Encoding header to chunked.


Valid Values: STANDARD | STREAMING

Request Body

The request accepts the following binary data.

Body (p. 48)

The bytes to be stored.

Response Syntax

```plaintext
HTTP/1.1 200
Content-type: application/json

{  
  "ContentSHA256": "string",
  "ETag": "string",
  "StorageClass": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ContentSHA256 (p. 49)

The SHA256 digest of the object that is persisted.

Type: String

Length Constraints: Fixed length of 64.

Pattern: [0-9A-Fa-f]{64}
**ETag (p. 49)**

Unique identifier of the object in the container.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: [0-9A-Fa-f]+

**StorageClass (p. 49)**

The storage class where the object was persisted. The class should be “Temporal”.

Type: String


Valid Values: STANDARD | TEMPORAL

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 61).

**ContainerNotFoundException**

The specified container was not found for the specified account.

HTTP Status Code: 404

**InternalServer>Error**

The service is temporarily unavailable.

HTTP Status Code: 500

**Example**

**Regular Upload Request**

The following request puts a file to the container that is identified by the endpoint that is specified in the Host: header. It posts it to the folder premium/canada in that container and names the file mlaw.avi.

```
POST premium/canada/mlaw.avi
Host: aaabbcccddee.files.mediastore-us-west-2.com
x-amz-date: 20170323T120000Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIAIOSFODNN7EXAMPLE/20141123/us-west-2/mediastore/aws4_request,SignedHeaders=host;x-amz-date;x-amz-mediastore-version,Signature=9257c16da6b25a715ce900a5b45b03da0447acfd430195dcb540091b12966f2a2
Content-Length: 0
x-amz-mediastore-version: 2016-07-11
```

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Go - Pilot
• AWS SDK for Java
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS SDK for Ruby V2
Data Types

The following data types are supported by AWS Elemental MediaStore:

- Container (p. 53)
- CorsRule (p. 55)

The following data types are supported by AWS Elemental MediaStore Data Plane:

- Item (p. 57)

AWS Elemental MediaStore

The following data types are supported by AWS Elemental MediaStore:

- Container (p. 53)
- CorsRule (p. 55)
Container
Service: AWS Elemental MediaStore

This section describes operations that you can perform on an AWS Elemental MediaStore container.

Contents

AccessLoggingEnabled

The state of access logging on the container. This value is `false` by default, indicating that AWS Elemental MediaStore does not send access logs to Amazon CloudWatch Logs. When you enable access logging on the container, MediaStore changes this value to `true`, indicating that the service delivers access logs for objects stored in that container to CloudWatch Logs.

Type: Boolean
Required: No

ARN

The Amazon Resource Name (ARN) of the container. The ARN has the following format:

arn:aws:<region>:<account that owns this container>:container/<name of container>


Type: String
Pattern: arn:aws:mediastore:[a-z]+-[a-z]+--\d:\d\{12\}:container/\w{1,255}

Required: No

CreationTime

Unix timestamp.

Type: Timestamp
Required: No

Endpoint

The DNS endpoint of the container. Use the endpoint to identify the specific container when sending requests to the data plane. The service assigns this value when the container is created. Once the value has been assigned, it does not change.

Type: String
Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Name

The name of the container.

Type: String
Length Constraints: Minimum length of 1. Maximum length of 255.
Pattern: [\w-]+
Required: No

**Status**

The status of container creation or deletion. The status is one of the following: CREATING, ACTIVE, or DELETING. While the service is creating the container, the status is CREATING. When the endpoint is available, the status changes to ACTIVE.

Type: String


Valid Values: ACTIVE | CREATING | DELETING

Required: No

**See Also**

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2
CorsRule

Service: AWS Elemental MediaStore

A rule for a CORS policy. You can add up to 100 rules to a CORS policy. If more than one rule applies, the service uses the first applicable rule listed.

Contents

AllowedHeaders

Specifies which headers are allowed in a preflight OPTIONS request through the Access-Control-Request-Headers header. Each header name that is specified in Access-Control-Request-Headers must have a corresponding entry in the rule. Only the headers that were requested are sent back.

This element can contain only one wildcard character (*).

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.


Pattern: \[\u0009\u000A\u000D\u0020-\u00FF]+

Required: Yes

AllowedMethods

Identifies an HTTP method that the origin that is specified in the rule is allowed to execute.

Each CORS rule must contain at least one AllowedMethods and one AllowedOrigins element.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 4 items.

Valid Values: PUT | GET | DELETE | HEAD

Required: No

AllowedOrigins

One or more response headers that you want users to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).

Each CORS rule must have at least one AllowedOrigins element. The string value can include only one wildcard character (*), for example, http://*.example.com. Additionally, you can specify only one wildcard character to allow cross-origin access for all origins.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Length Constraints: Minimum length of 1. Maximum length of 2048.

Pattern: \[\u0009\u000A\u000D\u0020-\u00FF]+

Required: Yes

ExposeHeaders

One or more headers in the response that you want users to be able to access from their applications (for example, from a JavaScript XMLHttpRequest object).
This element is optional for each rule.

Type: Array of strings

Array Members: Minimum number of 0 items. Maximum number of 100 items.


Pattern: \[\\u0009\\u000A\\u000D\\u0020-\u00FF]+

Required: No

MaxAgeSeconds

The time in seconds that your browser caches the preflight response for the specified resource.

A CORS rule can have only one MaxAgeSeconds element.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2

AWS Elemental MediaStore Data Plane

The following data types are supported by AWS Elemental MediaStore Data Plane:

- Item (p. 57)
Item

Service: AWS Elemental MediaStore Data Plane

A metadata entry for a folder or object.

Contents

**ContentLength**

The length of the item in bytes.
Type: Long
Valid Range: Minimum value of 0.
Required: No

**ContentType**

The content type of the item.
Type: String
Pattern: ^[\w\-\_/\.\+]{1,255}$
Required: No

**ETag**

The ETag that represents a unique instance of the item.
Type: String
Length Constraints: Minimum length of 1. Maximum length of 64.
Pattern: [0-9A-Za-z]+
Required: No

**LastModified**

The date and time that the item was last modified.
Type: Timestamp
Required: No

**Name**

The name of the item.
Type: String
Pattern: [A-Za-z0-9_\.\-\_]+
Required: No

**Type**

The item type (folder or object).
Type: String
Valid Values: OBJECT | FOLDER
Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Go - Pilot
- AWS SDK for Java
- AWS SDK for Ruby V2
Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see Signature Version 4 Signing Process in the Amazon Web Services General Reference.

**Action**

The action to be performed.

Type: string

Required: Yes

**Version**

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

**X-Amz-Algorithm**

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

**X-Amz-Credential**

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string (“aws4_request”). The value is expressed in the following format: access_key/YYYYMMDD/region/service/aws4_request.

For more information, see Task 2: Create a String to Sign for Signature Version 4 in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

**X-Amz-Date**

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'T'HHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is
not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the Amazon Web Services General Reference.

Type: string
Required: Conditional

**X-Amz-Security-Token**

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to AWS Services That Work with IAM in the IAM User Guide.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string
Required: Conditional

**X-Amz-Signature**

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Required: Conditional

**X-Amz-SignedHeaders**

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see Task 1: Create a Canonical Request For Signature Version 4 in the Amazon Web Services General Reference.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string
Required: Conditional
Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

**AccessDeniedException**
You do not have sufficient access to perform this action.

HTTP Status Code: 400

**IncompleteSignature**
The request signature does not conform to AWS standards.

HTTP Status Code: 400

**InternalFailure**
The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

**InvalidAction**
The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

**InvalidClientTokenId**
The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

**InvalidParameterCombination**
Parameters that must not be used together were used together.

HTTP Status Code: 400

**InvalidParameterValue**
An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

**InvalidQueryParameter**
The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

**MalformedQueryString**
The query string contains a syntax error.

HTTP Status Code: 404

**MissingAction**
The request is missing an action or a required parameter.

HTTP Status Code: 400
**MissingAuthenticationToken**

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

**MissingParameter**

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

**OptInRequired**

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

**RequestExpired**

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

**ServiceUnavailable**

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

**ThrottlingException**

The request was denied due to request throttling.

HTTP Status Code: 400

**ValidationError**

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400