

# Amazon Elastic Container Registry



# Amazon Elastic Container Registry: API Reference

# Table of Contents

<b>Welcome</b> .....	<b>1</b>
<b>Actions</b> .....	<b>2</b>
BatchCheckLayerAvailability .....	4
Request Syntax .....	4
Request Parameters .....	4
Response Syntax .....	5
Response Elements .....	6
Errors .....	6
Examples .....	7
See Also .....	8
BatchDeleteImage .....	9
Request Syntax .....	9
Request Parameters .....	9
Response Syntax .....	10
Response Elements .....	11
Errors .....	11
Examples .....	12
See Also .....	14
BatchGetImage .....	16
Request Syntax .....	16
Request Parameters .....	16
Response Syntax .....	17
Response Elements .....	18
Errors .....	18
Examples .....	19
See Also .....	21
BatchGetRepositoryScanningConfiguration .....	22
Request Syntax .....	22
Request Parameters .....	22
Response Syntax .....	22
Response Elements .....	23
Errors .....	23
See Also .....	24
CompleteLayerUpload .....	25

---

Request Syntax .....	25
Request Parameters .....	25
Response Syntax .....	26
Response Elements .....	27
Errors .....	27
See Also .....	29
CreatePullThroughCacheRule .....	30
Request Syntax .....	30
Request Parameters .....	30
Response Syntax .....	32
Response Elements .....	32
Errors .....	33
Examples .....	34
See Also .....	37
CreateRepository .....	38
Request Syntax .....	38
Request Parameters .....	38
Response Syntax .....	40
Response Elements .....	40
Errors .....	41
Examples .....	42
See Also .....	43
DeleteLifecyclePolicy .....	44
Request Syntax .....	44
Request Parameters .....	44
Response Syntax .....	44
Response Elements .....	45
Errors .....	46
Examples .....	46
See Also .....	48
DeletePullThroughCacheRule .....	49
Request Syntax .....	49
Request Parameters .....	49
Response Syntax .....	49
Response Elements .....	50
Errors .....	51

---

See Also .....	51
DeleteRegistryPolicy .....	53
Response Syntax .....	53
Response Elements .....	53
Errors .....	53
See Also .....	54
DeleteRepository .....	55
Request Syntax .....	55
Request Parameters .....	55
Response Syntax .....	56
Response Elements .....	56
Errors .....	57
Examples .....	57
See Also .....	59
DeleteRepositoryPolicy .....	60
Request Syntax .....	60
Request Parameters .....	60
Response Syntax .....	60
Response Elements .....	61
Errors .....	61
Examples .....	62
See Also .....	63
DescribeImageReplicationStatus .....	65
Request Syntax .....	65
Request Parameters .....	65
Response Syntax .....	66
Response Elements .....	66
Errors .....	67
See Also .....	68
DescribeImages .....	69
Request Syntax .....	69
Request Parameters .....	69
Response Syntax .....	71
Response Elements .....	72
Errors .....	72
Examples .....	73

---

See Also .....	75
DescribeImageScanFindings .....	77
Request Syntax .....	77
Request Parameters .....	77
Response Syntax .....	78
Response Elements .....	81
Errors .....	82
Examples .....	83
See Also .....	86
DescribePullThroughCacheRules .....	87
Request Syntax .....	87
Request Parameters .....	87
Response Syntax .....	88
Response Elements .....	89
Errors .....	89
See Also .....	90
DescribeRegistry .....	91
Response Syntax .....	91
Response Elements .....	91
Errors .....	92
See Also .....	92
DescribeRepositories .....	94
Request Syntax .....	94
Request Parameters .....	94
Response Syntax .....	95
Response Elements .....	96
Errors .....	96
Examples .....	97
See Also .....	98
GetAuthorizationToken .....	100
Request Syntax .....	100
Request Parameters .....	100
Response Syntax .....	100
Response Elements .....	101
Errors .....	101
Examples .....	102

---

See Also .....	103
GetDownloadUrlForLayer .....	104
Request Syntax .....	104
Request Parameters .....	104
Response Syntax .....	105
Response Elements .....	105
Errors .....	106
See Also .....	107
GetLifecyclePolicy .....	108
Request Syntax .....	108
Request Parameters .....	108
Response Syntax .....	108
Response Elements .....	109
Errors .....	110
Examples .....	110
See Also .....	112
GetLifecyclePolicyPreview .....	113
Request Syntax .....	113
Request Parameters .....	113
Response Syntax .....	115
Response Elements .....	115
Errors .....	117
Examples .....	118
See Also .....	119
GetRegistryPolicy .....	120
Response Syntax .....	120
Response Elements .....	120
Errors .....	120
See Also .....	121
GetRegistryScanningConfiguration .....	122
Response Syntax .....	122
Response Elements .....	122
Errors .....	123
See Also .....	123
GetRepositoryPolicy .....	124
Request Syntax .....	124

Request Parameters .....	124
Response Syntax .....	124
Response Elements .....	125
Errors .....	125
Examples .....	126
See Also .....	127
InitiateLayerUpload .....	129
Request Syntax .....	129
Request Parameters .....	129
Response Syntax .....	130
Response Elements .....	130
Errors .....	130
See Also .....	131
ListImages .....	132
Request Syntax .....	132
Request Parameters .....	132
Response Syntax .....	134
Response Elements .....	134
Errors .....	134
Examples .....	135
See Also .....	136
ListTagsForResource .....	138
Request Syntax .....	138
Request Parameters .....	138
Response Syntax .....	138
Response Elements .....	138
Errors .....	139
Examples .....	139
See Also .....	141
PutImage .....	142
Request Syntax .....	142
Request Parameters .....	142
Response Syntax .....	144
Response Elements .....	144
Errors .....	144
Examples .....	146



---

See Also .....	147
PutImageScanningConfiguration .....	149
Request Syntax .....	149
Request Parameters .....	149
Response Syntax .....	150
Response Elements .....	150
Errors .....	151
See Also .....	151
PutImageTagMutability .....	153
Request Syntax .....	153
Request Parameters .....	153
Response Syntax .....	154
Response Elements .....	154
Errors .....	155
Examples .....	155
See Also .....	156
PutLifecyclePolicy .....	158
Request Syntax .....	158
Request Parameters .....	158
Response Syntax .....	159
Response Elements .....	159
Errors .....	160
Examples .....	160
See Also .....	162
PutRegistryPolicy .....	163
Request Syntax .....	163
Request Parameters .....	163
Response Syntax .....	163
Response Elements .....	164
Errors .....	164
See Also .....	165
PutRegistryScanningConfiguration .....	166
Request Syntax .....	166
Request Parameters .....	166
Response Syntax .....	167
Response Elements .....	167

---

Errors .....	168
See Also .....	168
PutReplicationConfiguration .....	169
Request Syntax .....	169
Request Parameters .....	170
Response Syntax .....	170
Response Elements .....	170
Errors .....	171
See Also .....	171
SetRepositoryPolicy .....	173
Request Syntax .....	173
Request Parameters .....	173
Response Syntax .....	174
Response Elements .....	174
Errors .....	175
Examples .....	176
See Also .....	177
StartImageScan .....	178
Request Syntax .....	178
Request Parameters .....	178
Response Syntax .....	179
Response Elements .....	179
Errors .....	180
Examples .....	181
See Also .....	182
StartLifecyclePolicyPreview .....	184
Request Syntax .....	184
Request Parameters .....	184
Response Syntax .....	185
Response Elements .....	185
Errors .....	186
Examples .....	187
See Also .....	188
TagResource .....	190
Request Syntax .....	190
Request Parameters .....	190

---

Response Elements .....	191
Errors .....	191
See Also .....	191
UntagResource .....	193
Request Syntax .....	193
Request Parameters .....	193
Response Elements .....	193
Errors .....	193
See Also .....	194
UpdatePullThroughCacheRule .....	196
Request Syntax .....	196
Request Parameters .....	196
Response Syntax .....	197
Response Elements .....	197
Errors .....	198
See Also .....	199
UploadLayerPart .....	200
Request Syntax .....	200
Request Parameters .....	200
Response Syntax .....	202
Response Elements .....	202
Errors .....	203
See Also .....	204
ValidatePullThroughCacheRule .....	205
Request Syntax .....	205
Request Parameters .....	205
Response Syntax .....	206
Response Elements .....	206
Errors .....	207
See Also .....	208
<b>Data Types .....</b>	<b>209</b>
Attribute .....	211
Contents .....	211
See Also .....	211
AuthorizationData .....	212
Contents .....	212

---

See Also .....	212
AwsEcrContainerImageDetails .....	214
Contents .....	214
See Also .....	215
CvssScore .....	216
Contents .....	216
See Also .....	216
CvssScoreAdjustment .....	218
Contents .....	218
See Also .....	218
CvssScoreDetails .....	219
Contents .....	219
See Also .....	220
DescribeImagesFilter .....	221
Contents .....	221
See Also .....	221
EncryptionConfiguration .....	222
Contents .....	222
See Also .....	223
EnhancedImageScanFinding .....	224
Contents .....	224
See Also .....	226
Image .....	227
Contents .....	227
See Also .....	228
ImageDetail .....	229
Contents .....	229
See Also .....	231
ImageFailure .....	232
Contents .....	232
See Also .....	232
ImageIdentifier .....	234
Contents .....	234
See Also .....	234
ImageReplicationStatus .....	235
Contents .....	235

---

See Also .....	236
ImageScanFinding .....	237
Contents .....	237
See Also .....	238
ImageScanFindings .....	239
Contents .....	239
See Also .....	240
ImageScanFindingsSummary .....	241
Contents .....	241
See Also .....	241
ImageScanningConfiguration .....	243
Contents .....	243
See Also .....	243
ImageScanStatus .....	244
Contents .....	244
See Also .....	244
Layer .....	245
Contents .....	245
See Also .....	246
LayerFailure .....	247
Contents .....	247
See Also .....	247
LifecyclePolicyPreviewFilter .....	249
Contents .....	249
See Also .....	249
LifecyclePolicyPreviewResult .....	250
Contents .....	250
See Also .....	251
LifecyclePolicyPreviewSummary .....	252
Contents .....	252
See Also .....	252
LifecyclePolicyRuleAction .....	253
Contents .....	253
See Also .....	253
ListImagesFilter .....	254
Contents .....	254

---

See Also .....	254
PackageVulnerabilityDetails .....	255
Contents .....	255
See Also .....	256
PullThroughCacheRule .....	258
Contents .....	258
See Also .....	259
Recommendation .....	260
Contents .....	260
See Also .....	260
RegistryScanningConfiguration .....	261
Contents .....	261
See Also .....	261
RegistryScanningRule .....	262
Contents .....	262
See Also .....	262
Remediation .....	263
Contents .....	263
See Also .....	263
ReplicationConfiguration .....	264
Contents .....	264
See Also .....	264
ReplicationDestination .....	265
Contents .....	265
See Also .....	265
ReplicationRule .....	266
Contents .....	266
See Also .....	266
Repository .....	267
Contents .....	267
See Also .....	268
RepositoryFilter .....	270
Contents .....	270
See Also .....	270
RepositoryScanningConfiguration .....	272
Contents .....	272

---

See Also .....	273
RepositoryScanningConfigurationFailure .....	274
Contents .....	274
See Also .....	274
Resource .....	276
Contents .....	276
See Also .....	276
ResourceDetails .....	278
Contents .....	278
See Also .....	278
ScanningRepositoryFilter .....	279
Contents .....	279
See Also .....	279
ScoreDetails .....	280
Contents .....	280
See Also .....	280
Tag .....	281
Contents .....	281
See Also .....	281
VulnerablePackage .....	282
Contents .....	282
See Also .....	283
<b>Common Parameters .....</b>	<b>284</b>
<b>Common Errors .....</b>	<b>287</b>

# Welcome

Amazon Elastic Container Registry (Amazon ECR) is a managed container image registry service. Customers can use the familiar Docker CLI, or their preferred client, to push, pull, and manage images. Amazon ECR provides a secure, scalable, and reliable registry for your Docker or Open Container Initiative (OCI) images. Amazon ECR supports private repositories with resource-based permissions using IAM so that specific users or Amazon EC2 instances can access repositories and images.

Amazon ECR has service endpoints in each supported Region. For more information, see [Amazon ECR endpoints](#) in the *Amazon General Reference*.

This document was last published on April 25, 2024.



# Actions

The following actions are supported:

- [BatchCheckLayerAvailability](#)
- [BatchDeleteImage](#)
- [BatchGetImage](#)
- [BatchGetRepositoryScanningConfiguration](#)
- [CompleteLayerUpload](#)
- [CreatePullThroughCacheRule](#)
- [CreateRepository](#)
- [DeleteLifecyclePolicy](#)
- [DeletePullThroughCacheRule](#)
- [DeleteRegistryPolicy](#)
- [DeleteRepository](#)
- [DeleteRepositoryPolicy](#)
- [DescribeImageReplicationStatus](#)
- [DescribeImages](#)
- [DescribeImageScanFindings](#)
- [DescribePullThroughCacheRules](#)
- [DescribeRegistry](#)
- [DescribeRepositories](#)
- [GetAuthorizationToken](#)
- [GetDownloadUrlForLayer](#)
- [GetLifecyclePolicy](#)
- [GetLifecyclePolicyPreview](#)
- [GetRegistryPolicy](#)
- [GetRegistryScanningConfiguration](#)
- [GetRepositoryPolicy](#)
- [InitiateLayerUpload](#)
- [ListImages](#)

- [ListTagsForResource](#)
- [PutImage](#)
- [PutImageScanningConfiguration](#)
- [PutImageTagMutability](#)
- [PutLifecyclePolicy](#)
- [PutRegistryPolicy](#)
- [PutRegistryScanningConfiguration](#)
- [PutReplicationConfiguration](#)
- [SetRepositoryPolicy](#)
- [StartImageScan](#)
- [StartLifecyclePolicyPreview](#)
- [TagResource](#)
- [UntagResource](#)
- [UpdatePullThroughCacheRule](#)
- [UploadLayerPart](#)
- [ValidatePullThroughCacheRule](#)

# BatchCheckLayerAvailability

Checks the availability of one or more image layers in a repository.

When an image is pushed to a repository, each image layer is checked to verify if it has been uploaded before. If it has been uploaded, then the image layer is skipped.

## Note

This operation is used by the Amazon ECR proxy and is not generally used by customers for pulling and pushing images. In most cases, you should use the `docker` CLI to pull, tag, and push images.

## Request Syntax

```
{
  "layerDigests": [ "string" ],
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### layerDigests

The digests of the image layers to check.

Type: Array of strings

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

Length Constraints: Minimum length of 0. Maximum length of 1000.

Required: Yes

## registryId

The Amazon account ID associated with the registry that contains the image layers to check. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

## repositoryName

The name of the repository that is associated with the image layers to check.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
  "failures": [
    {
      "failureCode": "string",
      "failureReason": "string",
      "layerDigest": "string"
    }
  ],
  "layers": [
    {
      "layerAvailability": "string",
      "layerDigest": "string",
      "layerSize": number,
      "mediaType": "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.

### failures

Any failures associated with the call.

Type: Array of [LayerFailure](#) objects

### layers

A list of image layer objects corresponding to the image layer references in the request.

Type: Array of [Layer](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### Example

This example checks the availability of an image layer in the amazonlinux repository.

#### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 126
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.BatchCheckLayerAvailability
X-Amz-Date: 20161216T195733Zc
User-Agent: aws-cli/1.11.22 Python/2.7.12 Darwin/16.3.0 botocore/1.4.79
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "layerDigests": [
    "sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8"
  ],
  "repositoryName": "amazonlinux"
}
```

#### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 16 Dec 2016 19:57:33 GMT
Content-Type: application/x-amz-json-1.1
```

```
Content-Length: 233
Connection: keep-alive
x-amzn-RequestId: e2422faf-c3c9-11e6-a3ee-63b3b5dcf3b9

{
  "failures": [],
  "layers": [
    {
      "layerAvailability": "AVAILABLE",
      "layerDigest":
"sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8",
      "layerSize": 91768077,
      "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip"
    }
  ]
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# BatchDeleteImage

Deletes a list of specified images within a repository. Images are specified with either an `imageTag` or `imageDigest`.

You can remove a tag from an image by specifying the image's tag in your request. When you remove the last tag from an image, the image is deleted from your repository.

You can completely delete an image (and all of its tags) by specifying the image's digest in your request.

## Request Syntax

```
{
  "imageIds": [
    {
      "imageDigest": "string",
      "imageTag": "string"
    }
  ],
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### imageIds

A list of image ID references that correspond to images to delete. The format of the `imageIds` reference is `imageTag=tag` or `imageDigest=digest`.

Type: Array of [ImageIdentifier](#) objects

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

Required: Yes



## registryId

The Amazon account ID associated with the registry that contains the image to delete. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

## repositoryName

The repository that contains the image to delete.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "failures": [
    {
      "failureCode": "string",
      "failureReason": "string",
      "imageId": {
        "imageDigest": "string",
        "imageTag": "string"
      }
    }
  ],
  "imageIds": [
    {
      "imageDigest": "string",
      "imageTag": "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.

### failures

Any failures associated with the call.

Type: Array of [ImageFailure](#) objects

### imageIds

The image IDs of the deleted images.

Type: Array of [ImageIdentifier](#) 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](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### Example

This example deletes an image in the ubuntu repository with the imageTag value of xenial.

#### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 66
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.BatchDeleteImage
X-Amz-Date: 20161216T193711Z
User-Agent: aws-cli/1.11.22 Python/2.7.12 Darwin/16.3.0 botocore/1.4.79
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "repositoryName": "ubuntu",
  "imageIds": [
    {
      "imageTag": "xenial"
    }
  ]
}
```

#### Sample Response

```
HTTP/1.1 200 OK
Server: Server
```

```
Date: Fri, 16 Dec 2016 19:37:11 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 138
Connection: keep-alive
x-amzn-RequestId: 09cc7023-c3c7-11e6-8acf-61b7dd8abe56

{
  "failures": [],
  "imageIds": [
    {
      "imageDigest":
"sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ec8250899c397a",
      "imageTag": "xenial"
    }
  ]
}
```

## Example

This example deletes an image (and all of its tags) in the ubuntu repository with the `imageDigest` value of `sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ec8250899c397a`.

## Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 134
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.BatchDeleteImage
X-Amz-Date: 20161216T194250Z
User-Agent: aws-cli/1.11.22 Python/2.7.12 Darwin/16.3.0 botocore/1.4.79
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "repositoryName": "ubuntu",
  "imageIds": [
    {
      "imageDigest":
"sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ec8250899c397a"
    }
  ]
}
```

```
}
```

## Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 16 Dec 2016 19:42:50 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 248
Connection: keep-alive
x-amzn-RequestId: d441a9f6-c3c7-11e6-8acf-61b7dd8abe56

{
  "failures": [],
  "imageIds": [
    {
      "imageDigest":
"sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ec8250899c397a",
      "imageTag": "xenial"
    },
    {
      "imageDigest":
"sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ec8250899c397a",
      "imageTag": "latest"
    }
  ]
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)

- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# BatchGetImage

Gets detailed information for an image. Images are specified with either an `imageTag` or `imageDigest`.

When an image is pulled, the `BatchGetImage` API is called once to retrieve the image manifest.

## Request Syntax

```
{
  "acceptedMediaTypes": [ "string" ],
  "imageIds": [
    {
      "imageDigest": "string",
      "imageTag": "string"
    }
  ],
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### acceptedMediaTypes

The accepted media types for the request.

Valid values: `application/vnd.docker.distribution.manifest.v1+json` | `application/vnd.docker.distribution.manifest.v2+json` | `application/vnd.oci.image.manifest.v1+json`

Type: Array of strings

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

Required: No

## imageIds

A list of image ID references that correspond to images to describe. The format of the imageIds reference is imageTag=tag or imageDigest=digest.

Type: Array of [ImageIdentifier](#) objects

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

Required: Yes

## registryId

The Amazon account ID associated with the registry that contains the images to describe. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

## repositoryName

The repository that contains the images to describe.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
  "failures": [
    {
      "failureCode": "string",
      "failureReason": "string",
      "imageId": {
        "imageDigest": "string",
```



```
        "imageTag": "string"
      }
    },
    ],
    "images": [
      {
        "imageId": {
          "imageDigest": "string",
          "imageTag": "string"
        },
        "imageManifest": "string",
        "imageManifestMediaType": "string",
        "registryId": "string",
        "repositoryName": "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.

### failures

Any failures associated with the call.

Type: Array of [ImageFailure](#) objects

### images

A list of image objects corresponding to the image references in the request.

Type: Array of [Image](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **LimitExceededException**

The operation did not succeed because it would have exceeded a service limit for your account. For more information, see [Amazon ECR service quotas](#) in the Amazon Elastic Container Registry User Guide.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **UnableToGetUpstreamImageException**

The image or images were unable to be pulled using the pull through cache rule. This is usually caused because of an issue with the Secrets Manager secret containing the credentials for the upstream registry.

HTTP Status Code: 400

## **Examples**

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example describes an image in the `amazonlinux` repository with the `imageTag` value of `latest`.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 71
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.BatchGetImage
X-Amz-Date: 20161216T195356Z
User-Agent: aws-cli/1.11.22 Python/2.7.12 Darwin/16.3.0 botocore/1.4.79
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "repositoryName": "amazonlinux",
  "imageIds": [
    {
      "imageTag": "latest"
    }
  ]
}
```

### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 16 Dec 2016 19:53:56 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 800
Connection: keep-alive
x-amzn-RequestId: 60dc1ea1-c3c9-11e6-aa04-25c3a5fb1b54

{
  "failures": [],
  "images": [
    {
      "imageId": {
        "imageDigest":
          "sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807",
```

```
    "imageTag": "latest"
  },
  "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType\n\": \"application/vnd.docker.distribution.manifest.v2+json\n\",,\n  \"config\": {\n    \"mediaType\": \"application/\nvnd.docker.container.image.v1+json\",,\n    \"size\": 1486,\n    \"digest\":\n  \"sha256:5b52b314511a611975c2c65e695d920acdf8ae8848fe0ef00b7d018d1f118b64\"\n  },,\n  \"layers\": [\n    {\n      \"mediaType\": \"application/\nvnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 91768077,\n      \"digest\":\n  \"sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8\"\n    },,\n    ]\n  },\n  \"registryId\": \"012345678910\",,\n  \"repositoryName\": \"amazonlinux\"\n}\n]\n}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# BatchGetRepositoryScanningConfiguration

Gets the scanning configuration for one or more repositories.

## Request Syntax

```
{  
  "repositoryNames": [ "string" ]  
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### repositoryNames

One or more repository names to get the scanning configuration for.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 25 items.

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{  
  "failures": [  
    {  
      "failureCode": "string",  
      "failureReason": "string",  
      "repositoryName": "string"  
    }  
  ],  
}
```

```
"scanningConfigurations": [  
  {  
    "appliedScanFilters": [  
      {  
        "filter": "string",  
        "filterType": "string"  
      }  
    ],  
    "repositoryArn": "string",  
    "repositoryName": "string",  
    "scanFrequency": "string",  
    "scanOnPush": boolean  
  }  
]
```

## 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.

### failures

Any failures associated with the call.

Type: Array of [RepositoryScanningConfigurationFailure](#) objects

### scanningConfigurations

The scanning configuration for the requested repositories.

Type: Array of [RepositoryScanningConfiguration](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# CompleteLayerUpload

Notifies Amazon ECR that the image layer upload has completed for a specified registry, repository name, and upload ID. You can optionally provide a sha256 digest of the image layer for data validation purposes.

When an image is pushed, the CompleteLayerUpload API is called once per each new image layer to verify that the upload has completed.

## Note

This operation is used by the Amazon ECR proxy and is not generally used by customers for pulling and pushing images. In most cases, you should use the `docker` CLI to pull, tag, and push images.

## Request Syntax

```
{
  "layerDigests": [ "string" ],
  "registryId": "string",
  "repositoryName": "string",
  "uploadId": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### layerDigests

The sha256 digest of the image layer.

Type: Array of strings

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

Pattern: `[a-zA-Z0-9-_.]+:[a-fA-F0-9]+`



Required: Yes

### registryId

The Amazon account ID associated with the registry to which to upload layers. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

### repositoryName

The name of the repository to associate with the image layer.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

### uploadId

The upload ID from a previous [InitiateLayerUpload](#) operation to associate with the image layer.

Type: String

Pattern: `[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}`

Required: Yes

## Response Syntax

```
{
  "layerDigest": "string",
  "registryId": "string",
  "repositoryName": "string",
  "uploadId": "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.

### layerDigest

The sha256 digest of the image layer.

Type: String

Pattern: `[a-zA-Z0-9-_.]+:[a-fA-F0-9]+`

### registryId

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

### uploadId

The upload ID associated with the layer.

Type: String

Pattern: `[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

## **EmptyUploadException**

The specified layer upload does not contain any layer parts.

HTTP Status Code: 400

## **InvalidLayerException**

The layer digest calculation performed by Amazon ECR upon receipt of the image layer does not match the digest specified.

HTTP Status Code: 400

## **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## **KmsException**

The operation failed due to a KMS exception.

HTTP Status Code: 400

## **LayerAlreadyExistsException**

The image layer already exists in the associated repository.

HTTP Status Code: 400

## **LayerPartTooSmallException**

Layer parts must be at least 5 MiB in size.

HTTP Status Code: 400

## **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## UploadNotFoundException

The upload could not be found, or the specified upload ID is not valid for this repository.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# CreatePullThroughCacheRule

Creates a pull through cache rule. A pull through cache rule provides a way to cache images from an upstream registry source in your Amazon ECR private registry. For more information, see [Using pull through cache rules](#) in the *Amazon Elastic Container Registry User Guide*.

## Request Syntax

```
{
  "credentialArn": "string",
  "ecrRepositoryPrefix": "string",
  "registryId": "string",
  "upstreamRegistry": "string",
  "upstreamRegistryUrl": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [credentialArn](#)

The Amazon Resource Name (ARN) of the Amazon Secrets Manager secret that identifies the credentials to authenticate to the upstream registry.

Type: String

Length Constraints: Minimum length of 50. Maximum length of 612.

Pattern: `^arn:aws:secretsmanager:[a-zA-Z0-9-:]+:secret:ecr\-\pullthroughcache\[a-zA-Z0-9\|_+=.@-\]+`

Required: No

### [ecrRepositoryPrefix](#)

The repository name prefix to use when caching images from the source registry.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

### registryId

The Amazon account ID associated with the registry to create the pull through cache rule for. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

### upstreamRegistry

The name of the upstream registry.

Type: String

Valid Values: `ecr-public` | `quay` | `k8s` | `docker-hub` | `github-container-registry` | `azure-container-registry`

Required: No

### upstreamRegistryUrl

The registry URL of the upstream public registry to use as the source for the pull through cache rule. The following is the syntax to use for each supported upstream registry.

- Amazon ECR Public (`ecr-public`) - `public.ecr.aws`
- Docker Hub (`docker-hub`) - `registry-1.docker.io`
- Quay (`quay`) - `quay.io`
- Kubernetes (`k8s`) - `registry.k8s.io`
- GitHub Container Registry (`github-container-registry`) - `ghcr.io`
- Microsoft Azure Container Registry (`azure-container-registry`) - `<custom>.azurecr.io`

Type: String

Required: Yes

## Response Syntax

```
{
  "createdAt": number,
  "credentialArn": "string",
  "ecrRepositoryPrefix": "string",
  "registryId": "string",
  "upstreamRegistry": "string",
  "upstreamRegistryUrl": "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.

### createdAt

The date and time, in JavaScript date format, when the pull through cache rule was created.

Type: Timestamp

### credentialArn

The Amazon Resource Name (ARN) of the Amazon Secrets Manager secret associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 50. Maximum length of 612.

Pattern: `^arn:aws:secretsmanager:[a-zA-Z0-9-:]+:secret:ecr\-\pullthroughcache\[a-zA-Z0-9\|_+=.@-\]+`

### ecrRepositoryPrefix

The Amazon ECR repository prefix associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

## registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

## upstreamRegistry

The name of the upstream registry associated with the pull through cache rule.

Type: String

Valid Values: `ecr-public` | `quay` | `k8s` | `docker-hub` | `github-container-registry` | `azure-container-registry`

## upstreamRegistryUrl

The upstream registry URL associated with the pull through cache rule.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **LimitExceededException**

The operation did not succeed because it would have exceeded a service limit for your account. For more information, see [Amazon ECR service quotas](#) in the Amazon Elastic Container Registry User Guide.

HTTP Status Code: 400

### **PullThroughCacheRuleAlreadyExistsException**

A pull through cache rule with these settings already exists for the private registry.



HTTP Status Code: 400

### **SecretNotFoundException**

The ARN of the secret specified in the pull through cache rule was not found. Update the pull through cache rule with a valid secret ARN and try again.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **UnableToAccessSecretException**

The secret is unable to be accessed. Verify the resource permissions for the secret and try again.

HTTP Status Code: 400

### **UnableToDecryptSecretValueException**

The secret is accessible but is unable to be decrypted. Verify the resource permissions and try again.

HTTP Status Code: 400

### **UnsupportedUpstreamRegistryException**

The specified upstream registry isn't supported.

HTTP Status Code: 400

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## **Examples**

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example creates a pull through cache rule for an upstream registry that requires authentication.

### Sample Request

```
POST / HTTP/1.1
Host: api.ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.CreatePullThroughCacheRule
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/2.13.22 Python/3.11.5 Darwin/16.7.0 botocore/1.12.180
X-Amz-Date: 20231003T155747Z
Authorization: AUTHPARAMS
Content-Length: 268

{
  "ecrRepositoryPrefix": "docker-hub",
  "upstreamRegistryUrl": "registry-1.docker.io",
  "credentialArn": "arn:aws:secretsmanager:us-west-2:012345678910:secret:ecr-
pullthroughcache/example1234"
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 180
Connection: keep-alive

{
  "ecrRepositoryPrefix": "docker-hub",
  "upstreamRegistryUrl": "registry-1.docker.io",
  "createdAt": "2023-10-03T15:57:48.411000+00:00",
```

```
"registryId": "012345678910",
"upstreamRegistry": "docker-hub",
"credentialArn": "arn:aws:secretsmanager:us-west-2:012345678910:secret:ecr-
pullthroughcache/example1234"
}
```

## Example

This example creates a pull through cache rule for an upstream registry that does not require authentication.

## Sample Request

```
POST / HTTP/1.1
Host: api.ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.CreatePullThroughCacheRule
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/2.13.22 Python/3.11.5 Darwin/16.7.0 botocore/1.12.180
X-Amz-Date: 20231003T155747Z
Authorization: AUTHPARAMS
Content-Length: 268

{
  "ecrRepositoryPrefix": "ecr-public",
  "upstreamRegistryUrl": "public.ecr.aws"
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 180
Connection: keep-alive

{
  "ecrRepositoryPrefix": "ecr-public",
  "upstreamRegistryUrl": "public.ecr.aws",
  "createdAt": "2023-10-03T15:57:48.411000+00:00",
  "registryId": "012345678910",
  "upstreamRegistry": "ecr-public"
}
```

```
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# CreateRepository

Creates a repository. For more information, see [Amazon ECR repositories](#) in the *Amazon Elastic Container Registry User Guide*.

## Request Syntax

```
{
  "encryptionConfiguration": {
    "encryptionType": "string",
    "kmsKey": "string"
  },
  "imageScanningConfiguration": {
    "scanOnPush": boolean
  },
  "imageTagMutability": "string",
  "registryId": "string",
  "repositoryName": "string",
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [encryptionConfiguration](#)

The encryption configuration for the repository. This determines how the contents of your repository are encrypted at rest.

Type: [EncryptionConfiguration](#) object

Required: No

## imageScanningConfiguration

The image scanning configuration for the repository. This determines whether images are scanned for known vulnerabilities after being pushed to the repository.

Type: [ImageScanningConfiguration](#) object

Required: No

## imageTagMutability

The tag mutability setting for the repository. If this parameter is omitted, the default setting of `MUTABLE` will be used which will allow image tags to be overwritten. If `IMMUTABLE` is specified, all image tags within the repository will be immutable which will prevent them from being overwritten.

Type: String

Valid Values: `MUTABLE` | `IMMUTABLE`

Required: No

## registryId

The Amazon account ID associated with the registry to create the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

## repositoryName

The name to use for the repository. The repository name may be specified on its own (such as `nginx-web-app`) or it can be prepended with a namespace to group the repository into a category (such as `project-a/nginx-web-app`).

The repository name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, and forward slashes.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## tags

The metadata that you apply to the repository to help you categorize and organize them. Each tag consists of a key and an optional value, both of which you define. Tag keys can have a maximum character length of 128 characters, and tag values can have a maximum length of 256 characters.

Type: Array of [Tag](#) objects

Required: No

## Response Syntax

```
{
  "repository": {
    "createdAt": number,
    "encryptionConfiguration": {
      "encryptionType": "string",
      "kmsKey": "string"
    },
    "imageScanningConfiguration": {
      "scanOnPush": boolean
    },
    "imageTagMutability": "string",
    "registryId": "string",
    "repositoryArn": "string",
    "repositoryName": "string",
    "repositoryUri": "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.

## [repository](#)

The repository that was created.

Type: [Repository](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **InvalidTagParameterException**

An invalid parameter has been specified. Tag keys can have a maximum character length of 128 characters, and tag values can have a maximum length of 256 characters.

HTTP Status Code: 400

### **KmsException**

The operation failed due to a KMS exception.

HTTP Status Code: 400

### **LimitExceededException**

The operation did not succeed because it would have exceeded a service limit for your account. For more information, see [Amazon ECR service quotas](#) in the Amazon Elastic Container Registry User Guide.

HTTP Status Code: 400

### **RepositoryAlreadyExistsException**

The specified repository already exists in the specified registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.



HTTP Status Code: 500

## TooManyTagsException

The list of tags on the repository is over the limit. The maximum number of tags that can be applied to a repository is 50.

HTTP Status Code: 400

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example creates a repository called `sample-repo` in the default registry for an account.

## Sample Request

```
POST / HTTP/1.1
Host: api.ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.CreateRepository
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.16.190 Python/3.6.1 Darwin/16.7.0 botocore/1.12.180
X-Amz-Date: 20190715T204735Z
Authorization: AUTHPARAMS
Content-Length: 33

{
  "repositoryName": "sample-repo"
}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 339
Connection: keep-alive

{
  "repository":{
    "repositoryArn":"arn:aws:ecr:us-west-2:012345678910:repository/sample-repo",
    "registryId":"012345678910",
    "repositoryName":"sample-repo",
    "repositoryUri":"012345678910.dkr.ecr.us-west-2.amazonaws.com/sample-repo",
    "createdAt":1.563223656E9,
    "imageTagMutability":"MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": false
    }
  }
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DeleteLifecyclePolicy

Deletes the lifecycle policy associated with the specified repository.

## Request Syntax

```
{
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### registryId

The Amazon account ID associated with the registry that contains the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
```

```
"lastEvaluatedAt": number,  
"lifecyclePolicyText": "string",  
"registryId": "string",  
"repositoryName": "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.

### lastEvaluatedAt

The time stamp of the last time that the lifecycle policy was run.

Type: Timestamp

### lifecyclePolicyText

The JSON lifecycle policy text.

Type: String

Length Constraints: Minimum length of 100. Maximum length of 30720.

### registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### LifecyclePolicyNotFoundException

The lifecycle policy could not be found, and no policy is set to the repository.

HTTP Status Code: 400

### RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make

requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example deletes a lifecycle policy for a repository called `project-a/amazon-ecs-sample` in the default registry for an account.

## Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.DeleteLifecyclePolicy
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.11.144 Python/3.6.1 Darwin/16.6.0 botocore/1.7.2
X-Amz-Date: 20170901T223937Z
Authorization: AUTHPARAMS
Content-Length: 48

{
  "repositoryName": "project-a/amazon-ecs-sample",
}
```

## Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 01 Sep 2017 19:42:18 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 340
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "lastEvaluatedAt":1.504295007E9,
  "lifecyclePolicyText":{"rules":[{"rulePriority":1,"description":"Expire
images older than 14 days","selection":{"tagStatus":"untagged","countType":
"sinceImagePushed","countUnit":"days","countNumber":14},"action":{"type":
"expire"}}]}},
  "registryId":"012345678910",
}
```

```
"repositoryName": "project-a/amazon-ecs-sample"  
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DeletePullThroughCacheRule

Deletes a pull through cache rule.

## Request Syntax

```
{  
  "ecrRepositoryPrefix": "string",  
  "registryId": "string"  
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### ecrRepositoryPrefix

The Amazon ECR repository prefix associated with the pull through cache rule to delete.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

### registryId

The Amazon account ID associated with the registry that contains the pull through cache rule. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

## Response Syntax

```
{
```



```
"createdAt": number,  
"credentialArn": "string",  
"ecrRepositoryPrefix": "string",  
"registryId": "string",  
"upstreamRegistryUrl": "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.

### createdAt

The timestamp associated with the pull through cache rule.

Type: Timestamp

### credentialArn

The Amazon Resource Name (ARN) of the Amazon Secrets Manager secret associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 50. Maximum length of 612.

Pattern: `^arn:aws:secretsmanager:[a-zA-Z0-9-:]+:secret:ecr\-\pullthroughcache\/[a-zA-Z0-9\/_+=.@-]+$`

### ecrRepositoryPrefix

The Amazon ECR repository prefix associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

### registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

### upstreamRegistryUrl

The upstream registry URL associated with the pull through cache rule.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **PullThroughCacheRuleNotFoundException**

The pull through cache rule was not found. Specify a valid pull through cache rule and try again.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DeleteRegistryPolicy

Deletes the registry permissions policy.

## Response Syntax

```
{
  "policyText": "string",
  "registryId": "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.

### policyText

The contents of the registry permissions policy that was deleted.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

### registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## RegistryPolicyNotFoundException

The registry doesn't have an associated registry policy.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DeleteRepository

Deletes a repository. If the repository isn't empty, you must either delete the contents of the repository or use the `force` option to delete the repository and have Amazon ECR delete all of its contents on your behalf.

## Request Syntax

```
{  
  "force": boolean,  
  "registryId": "string",  
  "repositoryName": "string"  
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### force

If true, deleting the repository force deletes the contents of the repository. If false, the repository must be empty before attempting to delete it.

Type: Boolean

Required: No

### registryId

The Amazon account ID associated with the registry that contains the repository to delete. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository to delete.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "repository": {
    "createdAt": number,
    "encryptionConfiguration": {
      "encryptionType": "string",
      "kmsKey": "string"
    },
    "imageScanningConfiguration": {
      "scanOnPush": boolean
    },
    "imageTagMutability": "string",
    "registryId": "string",
    "repositoryArn": "string",
    "repositoryName": "string",
    "repositoryUri": "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.

### repository

The repository that was deleted.

Type: [Repository](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **KmsException**

The operation failed due to a KMS exception.

HTTP Status Code: 400

### **RepositoryNotEmptyException**

The specified repository contains images. To delete a repository that contains images, you must force the deletion with the `force` parameter.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make



requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example deletes a repository named `ubuntu` in the default registry for an account.

### Sample Request

```
POST / HTTP/1.1
Host: api.ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.DeleteRepository
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.16.190 Python/3.6.1 Darwin/16.7.0 botocore/1.12.180
X-Amz-Date: 20190715T205933Z
Authorization: AUTHPARAMS
Content-Length: 33

{
  "repositoryName": "sample-repo"
}
```

### Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 252
Connection: keep-alive

{
  "repository": {
    "createdAt": 1.563223656E9,
    "registryId": "012345678910",
    "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/sample-repo",
    "repositoryName": "sample-repo",
    "repositoryUri": "012345678910.dkr.ecr.us-west-2.amazonaws.com/sample-repo"
  }
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DeleteRepositoryPolicy

Deletes the repository policy associated with the specified repository.

## Request Syntax

```
{
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### registryId

The Amazon account ID associated with the registry that contains the repository policy to delete. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository that is associated with the repository policy to delete.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
```

```
"policyText": "string",  
"registryId": "string",  
"repositoryName": "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.

### policyText

The JSON repository policy that was deleted from the repository.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

### registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **RepositoryPolicyNotFoundException**

The specified repository and registry combination does not have an associated repository policy.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## **Examples**

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### **Example**

This example deletes the repository policy from the ubuntu repository.

### **Sample Request**

```
POST / HTTP/1.1
Host: ecr.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 28
```

```
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.DeleteRepositoryPolicy
X-Amz-Date: 20151215T003722Z
User-Agent: aws-cli/1.9.10 Python/2.7.10 Darwin/14.5.0 botocore/1.3.10
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "repositoryName": "ubuntu"
}
```

## Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Tue, 15 Dec 2015 00:37:22 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 301
Connection: keep-alive
x-amzn-RequestId: 01817918-a2c4-11e5-a19f-014c7a9aad99

{
  "policyText": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement\" : [ {\n    \"Sid\n\" : \"AllowPull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : \"*\",\n    \"Action\" : [ \"ecr:BatchGetImage\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}",
  "registryId": "012345678910",
  "repositoryName": "ubuntu"
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)

- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DescribeImageReplicationStatus

Returns the replication status for a specified image.

## Request Syntax

```
{
  "imageId": {
    "imageDigest": "string",
    "imageTag": "string"
  },
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### imageId

An object with identifying information for an image in an Amazon ECR repository.

Type: [ImageIdentifier](#) object

Required: Yes

### registryId

The Amazon account ID associated with the registry. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository that the image is in.



Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "imageId": {
    "imageDigest": "string",
    "imageTag": "string"
  },
  "replicationStatuses": [
    {
      "failureCode": "string",
      "region": "string",
      "registryId": "string",
      "status": "string"
    }
  ],
  "repositoryName": "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.

### imageId

An object with identifying information for an image in an Amazon ECR repository.

Type: [ImageIdentifier](#) object

### replicationStatuses

The replication status details for the images in the specified repository.

Type: Array of [ImageReplicationStatus](#) objects

## repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **ImageNotFoundException**

The image requested does not exist in the specified repository.

HTTP Status Code: 400

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DescribeImages

Returns metadata about the images in a repository.

## Note

Beginning with Docker version 1.9, the Docker client compresses image layers before pushing them to a V2 Docker registry. The output of the `docker images` command shows the uncompressed image size, so it may return a larger image size than the image sizes returned by [DescribeImages](#).

## Request Syntax

```
{
  "filter": {
    "tagStatus": "string"
  },
  "imageIds": [
    {
      "imageDigest": "string",
      "imageTag": "string"
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### filter

The filter key and value with which to filter your DescribeImages results.

Type: [DescribeImagesFilter](#) object

Required: No

### imageIds

The list of image IDs for the requested repository.

Type: Array of [ImageIdentifier](#) objects

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

Required: No

### maxResults

The maximum number of repository results returned by `DescribeImages` in paginated output. When this parameter is used, `DescribeImages` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeImages` request with the returned `nextToken` value. This value can be between 1 and 1000. If this parameter is not used, then `DescribeImages` returns up to 100 results and a `nextToken` value, if applicable. This option cannot be used when you specify images with `imageIds`.

Type: Integer

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

Required: No

### nextToken

The `nextToken` value returned from a previous paginated `DescribeImages` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return. This option cannot be used when you specify images with `imageIds`.

Type: String

Required: No

### registryId

The Amazon account ID associated with the registry that contains the repository in which to describe images. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The repository that contains the images to describe.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
  "imageDetails": [
    {
      "artifactMediaType": "string",
      "imageDigest": "string",
      "imageManifestMediaType": "string",
      "imagePushedAt": number,
      "imageScanFindingsSummary": {
        "findingSeverityCounts": {
          "string" : number
        },
        "imageScanCompletedAt": number,
        "vulnerabilitySourceUpdatedAt": number
      },
      "imageScanStatus": {
        "description": "string",
        "status": "string"
      },
      "imageSizeInBytes": number,
      "imageTags": [ "string" ],
      "lastRecordedPullTime": number,
      "registryId": "string",
      "repositoryName": "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.

### imageDetails

A list of [ImageDetail](#) objects that contain data about the image.

Type: Array of [ImageDetail](#) objects

### nextToken

The nextToken value to include in a future DescribeImages request. When the results of a DescribeImages request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **ImageNotFoundException**

The image requested does not exist in the specified repository.

HTTP Status Code: 400

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example describes the images in a repository named `hello-repository` in the default account. Note that the image with the digest `sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ecEXAMPLE` is tagged as `latest` and `tagtest`.

## Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 38
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.DescribeImages
X-Amz-Date: 20220301T194641Z
User-Agent: aws-cli/1.11.22 Python/2.7.12 Darwin/16.3.0 botocore/1.4.79
Content-Type: application/x-amz-json-1.1

Authorization: AUTHPARAMS

{
```



```
"repositoryName": "hello-repository"  
}
```

## Sample Response

```
HTTP/1.1 200 OK  
Server: Server  
Date: Fri, 16 Dec 2016 19:31:33 GMT  
Content-Type: application/x-amz-json-1.1  
Content-Length: 1107  
Connection: keep-alive  
x-amzn-RequestId: 404826b1-c3c6-11e6-a9e5-e3c203a2f07f  
  
{  
  "imageDetails": [  
    {  
      "registryId": "012345678910",  
      "repositoryName": "hello-repository",  
      "imageDigest":  
"sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ecEXAMPLE",  
      "imageTags": [  
        "tagtest"  
      ],  
      "imageSizeInBytes": 78447648,  
      "imagePushedAt": "2021-12-03T20:36:09+00:00",  
      "imageScanStatus": {  
        "status": "COMPLETE",  
        "description": "The scan was completed successfully."  
      },  
      "imageScanFindingsSummary": {  
        "imageScanCompletedAt": "2021-12-03T20:36:14+00:00",  
        "vulnerabilitySourceUpdatedAt": "2021-11-24T16:11:37+00:00",  
        "findingSeverityCounts": {  
          "HIGH": 4,  
          "MEDIUM": 76,  
          "INFORMATIONAL": 9,  
          "LOW": 50  
        }  
      },  
      "imageManifestMediaType": "application/  
vnd.docker.distribution.manifest.v2+json",
```

```
    "artifactMediaType": "application/vnd.docker.container.image.v1+json",
    "lastRecordedPullTime": "2022-02-23T05:06:01.514000+00:00"
  },
  {
    "registryId": "012345678910",
    "repositoryName": "hello-repository",
    "imageDigest":
"sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ecEXAMPLE",
    "imageTags": [
      "latest"
    ],
    "imageSizeInBytes": 78447648,
    "imagePushedAt": "2020-01-23T17:56:13+00:00",
    "imageScanStatus": {
      "status": "COMPLETE",
      "description": "The scan was completed successfully."
    },
    "imageScanFindingsSummary": {
      "imageScanCompletedAt": "2020-01-24T04:11:45+00:00",
      "vulnerabilitySourceUpdatedAt": "2020-01-23T20:25:17+00:00",
      "findingSeverityCounts": {
        "MEDIUM": 14,
        "INFORMATIONAL": 11,
        "LOW": 27
      }
    },
    "imageManifestMediaType": "application/
vnd.docker.distribution.manifest.v2+json",
    "lastRecordedPullTime": "2022-02-24T07:35:16.530000+00:00"
  }
]
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)

- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DescribeImageScanFindings

Returns the scan findings for the specified image.

## Request Syntax

```
{
  "imageId": {
    "imageDigest": "string",
    "imageTag": "string"
  },
  "maxResults": number,
  "nextToken": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### imageId

An object with identifying information for an image in an Amazon ECR repository.

Type: [ImageIdentifier](#) object

Required: Yes

### maxResults

The maximum number of image scan results returned by `DescribeImageScanFindings` in paginated output. When this parameter is used, `DescribeImageScanFindings` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeImageScanFindings` request with the returned `nextToken` value. This value can be between 1 and 1000. If this parameter is not used, then `DescribeImageScanFindings` returns up to 100 results and a `nextToken` value, if applicable.

Type: Integer

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

Required: No

### nextToken

The `nextToken` value returned from a previous paginated `DescribeImageScanFindings` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is null when there are no more results to return.

Type: String

Required: No

### registryId

The Amazon account ID associated with the registry that contains the repository in which to describe the image scan findings for. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

### repositoryName

The repository for the image for which to describe the scan findings.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "imageId": {
    "imageDigest": "string",
    "imageTag": "string"
  }
}
```

```
},
"imageScanFindings": {
  "enhancedFindings": [
    {
      "awsAccountId": "string",
      "description": "string",
      "findingArn": "string",
      "firstObservedAt": number,
      "lastObservedAt": number,
      "packageVulnerabilityDetails": {
        "cvss": [
          {
            "baseScore": number,
            "scoringVector": "string",
            "source": "string",
            "version": "string"
          }
        ],
        "referenceUrls": [ "string" ],
        "relatedVulnerabilities": [ "string" ],
        "source": "string",
        "sourceUrl": "string",
        "vendorCreatedAt": number,
        "vendorSeverity": "string",
        "vendorUpdatedAt": number,
        "vulnerabilityId": "string",
        "vulnerablePackages": [
          {
            "arch": "string",
            "epoch": number,
            "filePath": "string",
            "name": "string",
            "packageManager": "string",
            "release": "string",
            "sourceLayerHash": "string",
            "version": "string"
          }
        ]
      }
    },
    {
      "remediation": {
        "recommendation": {
          "text": "string",
          "url": "string"
        }
      }
    }
  ]
}
```

```
    },
    "resources": [
      {
        "details": {
          "awsEcrContainerImage": {
            "architecture": "string",
            "author": "string",
            "imageHash": "string",
            "imageTags": [ "string" ],
            "platform": "string",
            "pushedAt": number,
            "registry": "string",
            "repositoryName": "string"
          }
        },
        "id": "string",
        "tags": {
          "string" : "string"
        },
        "type": "string"
      }
    ],
    "score": number,
    "scoreDetails": {
      "cvss": {
        "adjustments": [
          {
            "metric": "string",
            "reason": "string"
          }
        ],
        "score": number,
        "scoreSource": "string",
        "scoringVector": "string",
        "version": "string"
      }
    },
    "severity": "string",
    "status": "string",
    "title": "string",
    "type": "string",
    "updatedAt": number
  }
],
```

```
"findings": [
  {
    "attributes": [
      {
        "key": "string",
        "value": "string"
      }
    ],
    "description": "string",
    "name": "string",
    "severity": "string",
    "uri": "string"
  }
],
"findingSeverityCounts": {
  "string" : number
},
"imageScanCompletedAt": number,
"vulnerabilitySourceUpdatedAt": number
},
"imageScanStatus": {
  "description": "string",
  "status": "string"
},
"nextToken": "string",
"registryId": "string",
"repositoryName": "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.

### [imageId](#)

An object with identifying information for an image in an Amazon ECR repository.

Type: [ImageIdentifier](#) object

### [imageScanFindings](#)

The information contained in the image scan findings.



Type: [ImageScanFindings](#) object

### [imageScanStatus](#)

The current state of the scan.

Type: [ImageScanStatus](#) object

### [nextToken](#)

The nextToken value to include in a future DescribeImageScanFindings request. When the results of a DescribeImageScanFindings request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

### [registryId](#)

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

### [repositoryName](#)

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **ImageNotFoundException**

The image requested does not exist in the specified repository.

HTTP Status Code: 400

## InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ScanNotFoundException

The specified image scan could not be found. Ensure that image scanning is enabled on the repository and try again.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example returns the image scan findings for an image using the image digest in a repository named `sample-repo` in the default registry for an account.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 141
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.DescribeImageScanFindings
X-Amz-Date: 20161216T201255Z
User-Agent: aws-cli/1.16.310 Python/3.6.1 Darwin/18.7.0 botocore/1.13.46
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS
```

```
{
  "repositoryName": "sample-repo",
  "imageId": {
    "imageDigest":
    "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
  }
}
```

### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 24 Jan 2020 03:48:07 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 33967
Connection: keep-alive
x-amzn-RequestId: 3081a92b-2066-41f8-8a47-0580288ada9e
```

```
{
  "imageScanFindings": {
    "findings": [
      {
        "name": "CVE-2019-5188",
        "description": "A code execution vulnerability exists in the directory rehashing functionality of E2fsprogs e2fsck 1.45.4. A specially crafted ext4 directory
```

```
can cause an out-of-bounds write on the stack, resulting in code execution. An
attacker can corrupt a partition to trigger this vulnerability.",
  "uri": "http://people.ubuntu.com/~ubuntu-security/cve/CVE-2019-5188",
  "severity": "MEDIUM",
  "attributes": [
    {
      "key": "package_version",
      "value": "1.44.1-1ubuntu1.1"
    },
    {
      "key": "package_name",
      "value": "e2fsprogs"
    },
    {
      "key": "CVSS2_VECTOR",
      "value": "AV:L/AC:L/Au:N/C:P/I:P/A:P"
    },
    {
      "key": "CVSS2_SCORE",
      "value": "4.6"
    }
  ]
},
"imageScanCompletedAt": 1579839105.0,
"vulnerabilitySourceUpdatedAt": 1579811117.0,
"findingSeverityCounts": {
  "MEDIUM": 1
}
},
"registryId": "012345678910",
"repositoryName": "sample-repo",
"imageId": {
  "imageDigest":
"sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
},
"imageScanStatus": {
  "status": "COMPLETE",
  "description": "The scan was completed successfully."
}
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DescribePullThroughCacheRules

Returns the pull through cache rules for a registry.

## Request Syntax

```
{
  "ecrRepositoryPrefixes": [ "string" ],
  "maxResults": number,
  "nextToken": "string",
  "registryId": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [ecrRepositoryPrefixes](#)

The Amazon ECR repository prefixes associated with the pull through cache rules to return. If no repository prefix value is specified, all pull through cache rules are returned.

Type: Array of strings

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

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

### [maxResults](#)

The maximum number of pull through cache rules returned by `DescribePullThroughCacheRulesRequest` in paginated output. When this parameter is used, `DescribePullThroughCacheRulesRequest` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `DescribePullThroughCacheRulesRequest` request

with the returned `nextToken` value. This value can be between 1 and 1000. If this parameter is not used, then `DescribePullThroughCacheRulesRequest` returns up to 100 results and a `nextToken` value, if applicable.

Type: Integer

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

Required: No

### nextToken

The `nextToken` value returned from a previous paginated `DescribePullThroughCacheRulesRequest` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is null when there are no more results to return.

Type: String

Required: No

### registryId

The Amazon account ID associated with the registry to return the pull through cache rules for. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

## Response Syntax

```
{
  "nextToken": "string",
  "pullThroughCacheRules": [
    {
      "createdAt": number,
      "credentialArn": "string",
      "ecrRepositoryPrefix": "string",
```

```
    "registryId": "string",
    "updatedAt": number,
    "upstreamRegistry": "string",
    "upstreamRegistryUrl": "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.

### nextToken

The nextToken value to include in a future DescribePullThroughCacheRulesRequest request. When the results of a DescribePullThroughCacheRulesRequest request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

### pullThroughCacheRules

The details of the pull through cache rules.

Type: Array of [PullThroughCacheRule](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **PullThroughCacheRuleNotFoundException**

The pull through cache rule was not found. Specify a valid pull through cache rule and try again.



HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## **See Also**

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DescribeRegistry

Describes the settings for a registry. The replication configuration for a repository can be created or updated with the [PutReplicationConfiguration](#) API action.

## Response Syntax

```
{
  "registryId": "string",
  "replicationConfiguration": {
    "rules": [
      {
        "destinations": [
          {
            "region": "string",
            "registryId": "string"
          }
        ],
        "repositoryFilters": [
          {
            "filter": "string",
            "filterType": "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.

### [registryId](#)

The ID of the registry.

Type: String

Pattern: [0-9]{12}

## [replicationConfiguration](#)

The replication configuration for the registry.

Type: [ReplicationConfiguration](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)

- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# DescribeRepositories

Describes image repositories in a registry.

## Request Syntax

```
{
  "maxResults": number,
  "nextToken": "string",
  "registryId": "string",
  "repositoryNames": [ "string" ]
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [maxResults](#)

The maximum number of repository results returned by `DescribeRepositories` in paginated output. When this parameter is used, `DescribeRepositories` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `DescribeRepositories` request with the returned `nextToken` value. This value can be between 1 and 1000. If this parameter is not used, then `DescribeRepositories` returns up to 100 results and a `nextToken` value, if applicable. This option cannot be used when you specify repositories with `repositoryNames`.

Type: Integer

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

Required: No

### [nextToken](#)

The `nextToken` value returned from a previous paginated `DescribeRepositories` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value

is `null` when there are no more results to return. This option cannot be used when you specify repositories with `repositoryNames`.

**Note**

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.

Type: String

Required: No

### registryId

The Amazon account ID associated with the registry that contains the repositories to be described. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

### repositoryNames

A list of repositories to describe. If this parameter is omitted, then all repositories in a registry are described.

Type: Array of strings

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

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

## Response Syntax

```
{  
  "nextToken": "string",
```

```
"repositories": [  
  {  
    "createdAt": number,  
    "encryptionConfiguration": {  
      "encryptionType": "string",  
      "kmsKey": "string"  
    },  
    "imageScanningConfiguration": {  
      "scanOnPush": boolean  
    },  
    "imageTagMutability": "string",  
    "registryId": "string",  
    "repositoryArn": "string",  
    "repositoryName": "string",  
    "repositoryUri": "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.

### [nextToken](#)

The nextToken value to include in a future DescribeRepositories request. When the results of a DescribeRepositories request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

### [repositories](#)

A list of repository objects corresponding to valid repositories.

Type: Array of [Repository](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

## InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example describes the repositories in the default registry for an account.

## Sample Request

```
POST / HTTP/1.1
Host: api.ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.DescribeRepositories
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.16.190 Python/3.6.1 Darwin/16.7.0 botocore/1.12.180
```



```
X-Amz-Date: 20190715T205400Z
Authorization: AUTHPARAMS
Content-Length: 2

{}
```

## Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f
Content-Type: application/x-amz-json-1.1
Content-Length: 1061
Connection: keep-alive

{
  "repositories": [
    {
      "createdAt": 1.563223656E9,
      "imageTagMutability": "MUTABLE",
      "registryId": "012345678910",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/sample-repo",
      "repositoryName": "sample-repo",
      "repositoryUri": "012345678910.dkr.ecr.us-west-2.amazonaws.com/sample-repo"
    },
    {
      "createdAt": 1.554824595E9,
      "imageTagMutability": "IMMUTABLE",
      "registryId": "012345678910",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/tagging-test",
      "repositoryName": "tagging-test",
      "repositoryUri": "012345678910.dkr.ecr.us-west-2.amazonaws.com/tagging-test"
    }
  ]
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)

- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# GetAuthorizationToken

Retrieves an authorization token. An authorization token represents your IAM authentication credentials and can be used to access any Amazon ECR registry that your IAM principal has access to. The authorization token is valid for 12 hours.

The `authorizationToken` returned is a base64 encoded string that can be decoded and used in a `docker login` command to authenticate to a registry. The Amazon CLI offers a `get-login-password` command that simplifies the login process. For more information, see [Registry authentication](#) in the *Amazon Elastic Container Registry User Guide*.

## Request Syntax

```
{
  "registryIds": [ "string" ]
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [registryIds](#)

*This parameter has been deprecated.*

A list of Amazon account IDs that are associated with the registries for which to get `AuthorizationData` objects. If you do not specify a registry, the default registry is assumed.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

Pattern: `[0-9]{12}`

Required: No

## Response Syntax

```
{
```

```
"authorizationData": [  
  {  
    "authorizationToken": "string",  
    "expiresAt": number,  
    "proxyEndpoint": "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.

### authorizationData

A list of authorization token data objects that correspond to the `registryIds` values in the request.

Type: Array of [AuthorizationData](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### Example

This example gets an authorization token for your default registry.

#### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.GetAuthorizationToken
X-Amz-Date: 20220516T185613Z
User-Agent: aws-cli/1.9.9 Python/2.7.10 Darwin/14.5.0 botocore/1.3.9
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{}
```

#### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Sun, 17 May 2022 06:56:13 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 1590
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
```

```
"authorizationData": [  
  {  
    "authorizationToken": "QVdT0kNpQzErSHF1ZXZPcUR...",  
    "expiresAt": "2022-05-17T06:56:13.652000+00:00",  
    "proxyEndpoint": "https://012345678910.dkr.ecr.us-east-1.amazonaws.com"  
  }  
]  
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# GetDownloadUrlForLayer

Retrieves the pre-signed Amazon S3 download URL corresponding to an image layer. You can only get URLs for image layers that are referenced in an image.

When an image is pulled, the `GetDownloadUrlForLayer` API is called once per image layer that is not already cached.

## Note

This operation is used by the Amazon ECR proxy and is not generally used by customers for pulling and pushing images. In most cases, you should use the `docker` CLI to pull, tag, and push images.

## Request Syntax

```
{
  "layerDigest": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### layerDigest

The digest of the image layer to download.

Type: String

Pattern: `[a-zA-Z0-9-_.]+:[a-fA-F0-9]+`

Required: Yes

## [registryId](#)

The Amazon account ID associated with the registry that contains the image layer to download. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

## [repositoryName](#)

The name of the repository that is associated with the image layer to download.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "downloadUrl": "string",
  "layerDigest": "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.

### [downloadUrl](#)

The pre-signed Amazon S3 download URL for the requested layer.

Type: String



## layerDigest

The digest of the image layer to download.

Type: String

Pattern: `[a-zA-Z0-9-_.]+:[a-fA-F0-9]+`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **LayerInaccessibleException**

The specified layer is not available because it is not associated with an image. Unassociated image layers may be cleaned up at any time.

HTTP Status Code: 400

### **LayersNotFoundException**

The specified layers could not be found, or the specified layer is not valid for this repository.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## UnableToGetUpstreamLayerException

There was an issue getting the upstream layer matching the pull through cache rule.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# GetLifecyclePolicy

Retrieves the lifecycle policy for the specified repository.

## Request Syntax

```
{
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### registryId

The Amazon account ID associated with the registry that contains the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
```

```
"lastEvaluatedAt": number,  
"lifecyclePolicyText": "string",  
"registryId": "string",  
"repositoryName": "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.

### lastEvaluatedAt

The time stamp of the last time that the lifecycle policy was run.

Type: Timestamp

### lifecyclePolicyText

The JSON lifecycle policy text.

Type: String

Length Constraints: Minimum length of 100. Maximum length of 30720.

### registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### LifecyclePolicyNotFoundException

The lifecycle policy could not be found, and no policy is set to the repository.

HTTP Status Code: 400

### RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make

requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example retrieves the lifecycle policy for a repository called `project-a/amazon-ecs-sample` in the default registry for an account.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.GetLifecyclePolicy
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.11.144 Python/3.6.1 Darwin/16.6.0 botocore/1.7.2
X-Amz-Date: 20170901T210647Z
Authorization: AUTHPARAMS
Content-Length: 48

{
  "repositoryName": "project-a/amazon-ecs-sample"
}
```

### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 01 Sep 2017 21:06:48 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 372
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "lastEvaluatedAt":1.504295007E9,
  "lifecyclePolicyText":{"rules":[{"rulePriority":1,"description":"Expire
images older than 14 days","selection":{"tagStatus":"untagged","countType":
"sinceImagePushed","countUnit":"days","countNumber":14},"action":{"type":
"expire"}}]}},
  "registryId":"012345678910",
```

```
"repositoryName": "project-a/amazon-ecs-sample"  
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# GetLifecyclePolicyPreview

Retrieves the results of the lifecycle policy preview request for the specified repository.

## Request Syntax

```
{
  "filter": {
    "tagStatus": "string"
  },
  "imageIds": [
    {
      "imageDigest": "string",
      "imageTag": "string"
    }
  ],
  "maxResults": number,
  "nextToken": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### filter

An optional parameter that filters results based on image tag status and all tags, if tagged.

Type: [LifecyclePolicyPreviewFilter](#) object

Required: No

### imageIds

The list of imageIDs to be included.

Type: Array of [ImageIdentifier](#) objects

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



Required: No

### maxResults

The maximum number of repository results returned by `GetLifecyclePolicyPreviewRequest` in paginated output. When this parameter is used, `GetLifecyclePolicyPreviewRequest` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `GetLifecyclePolicyPreviewRequest` request with the returned `nextToken` value. This value can be between 1 and 1000. If this parameter is not used, then `GetLifecyclePolicyPreviewRequest` returns up to 100 results and a `nextToken` value, if applicable. This option cannot be used when you specify images with `imageIds`.

Type: Integer

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

Required: No

### nextToken

The `nextToken` value returned from a previous paginated `GetLifecyclePolicyPreviewRequest` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return. This option cannot be used when you specify images with `imageIds`.

Type: String

Required: No

### registryId

The Amazon account ID associated with the registry that contains the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

### repositoryName

The name of the repository.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "lifecyclePolicyText": "string",
  "nextToken": "string",
  "previewResults": [
    {
      "action": {
        "type": "string"
      },
      "appliedRulePriority": number,
      "imageDigest": "string",
      "imagePushedAt": number,
      "imageTags": [ "string" ]
    }
  ],
  "registryId": "string",
  "repositoryName": "string",
  "status": "string",
  "summary": {
    "expiringImageTotalCount": 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.

## lifecyclePolicyText

The JSON lifecycle policy text.

Type: String

Length Constraints: Minimum length of 100. Maximum length of 30720.

## nextToken

The nextToken value to include in a future GetLifecyclePolicyPreview request. When the results of a GetLifecyclePolicyPreview request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

## previewResults

The results of the lifecycle policy preview request.

Type: Array of [LifecyclePolicyPreviewResult](#) objects

## registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

## repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## status

The status of the lifecycle policy preview request.

Type: String

Valid Values: IN\_PROGRESS | COMPLETE | EXPIRED | FAILED

### summary

The list of images that is returned as a result of the action.

Type: [LifecyclePolicyPreviewSummary](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **LifecyclePolicyPreviewNotFoundException**

There is no dry run for this repository.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### Example

This example retrieves the result of a lifecycle policy preview for a repository called `project-a/amazon-ecs-sample` in the default registry for an account.

#### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.GetLifecyclePolicyPreview
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.11.144 Python/3.6.1 Darwin/16.6.0 botocore/1.7.2
X-Amz-Date: 20170901T222304Z
Authorization: AUTHPARAMS
Content-Length: 48

{
  "repositoryName": "project-a/amazon-ecs-sample"
}
```

#### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 01 Sep 2017 22:23:06 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 640
Connection: keep-alive
```

```
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\n\": 1,\n      \"description\": \"Expire images older than 14 days\",\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\": \"days\",\n        \"countNumber\": 14\n      },\n      \"action\": {\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}",
  "previewResults": [],
  "registryId": "012345678910",
  "repositoryName": "project-a/amazon-ecs-sample",
  "status": "COMPLETE",
  "summary": {"expiringImageTotalCount": 0}
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# GetRegistryPolicy

Retrieves the permissions policy for a registry.

## Response Syntax

```
{
  "policyText": "string",
  "registryId": "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.

### policyText

The JSON text of the permissions policy for a registry.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

### registryId

The ID of the registry.

Type: String

Pattern: [0-9]{12}

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## RegistryPolicyNotFoundException

The registry doesn't have an associated registry policy.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)



# GetRegistryScanningConfiguration

Retrieves the scanning configuration for a registry.

## Response Syntax

```
{
  "registryId": "string",
  "scanningConfiguration": {
    "rules": [
      {
        "repositoryFilters": [
          {
            "filter": "string",
            "filterType": "string"
          }
        ],
        "scanFrequency": "string"
      }
    ],
    "scanType": "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.

### registryId

The ID of the registry.

Type: String

Pattern: [0-9]{12}

### scanningConfiguration

The scanning configuration for the registry.

Type: [RegistryScanningConfiguration](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# GetRepositoryPolicy

Retrieves the repository policy for the specified repository.

## Request Syntax

```
{
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### registryId

The Amazon account ID associated with the registry that contains the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository with the policy to retrieve.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
```

```
"policyText": "string",  
"registryId": "string",  
"repositoryName": "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.

### policyText

The JSON repository policy text associated with the repository.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

### registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **RepositoryPolicyNotFoundException**

The specified repository and registry combination does not have an associated repository policy.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## **Examples**

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### **Example**

This example gets the repository policy for the ubuntu repository.

### **Sample Request**

```
POST / HTTP/1.1
Host: ecr.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 28
```

```
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.GetRepositoryPolicy
X-Amz-Date: 20151215T002404Z
User-Agent: aws-cli/1.9.10 Python/2.7.10 Darwin/14.5.0 botocore/1.3.10
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "repositoryName": "ubuntu"
}
```

## Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Tue, 15 Dec 2015 00:24:04 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 301
Connection: keep-alive
x-amzn-RequestId: 25da0b72-a2c2-11e5-8543-ebda6fb1393b

{
  "policyText": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement\" : [ {\n    \"Sid\n\" : \"AllowPull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : \"*\",\n    \"Action\" : [ \"ecr:BatchGetImage\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}",
  "registryId": "012345678910",
  "repositoryName": "ubuntu"
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)

- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# InitiateLayerUpload

Notifies Amazon ECR that you intend to upload an image layer.

When an image is pushed, the InitiateLayerUpload API is called once per image layer that has not already been uploaded. Whether or not an image layer has been uploaded is determined by the BatchCheckLayerAvailability API action.

## Note

This operation is used by the Amazon ECR proxy and is not generally used by customers for pulling and pushing images. In most cases, you should use the docker CLI to pull, tag, and push images.

## Request Syntax

```
{
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### registryId

The Amazon account ID associated with the registry to which you intend to upload layers. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository to which you intend to upload layers.



Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "partSize": number,
  "uploadId": "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.

### partSize

The size, in bytes, that Amazon ECR expects future layer part uploads to be.

Type: Long

Valid Range: Minimum value of 0.

### uploadId

The upload ID for the layer upload. This parameter is passed to further [UploadLayerPart](#) and [CompleteLayerUpload](#) operations.

Type: String

Pattern: `[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

## InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## KmsException

The operation failed due to a KMS exception.

HTTP Status Code: 400

## RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# ListImages

Lists all the image IDs for the specified repository.

You can filter images based on whether or not they are tagged by using the `tagStatus` filter and specifying either `TAGGED`, `UNTAGGED` or `ANY`. For example, you can filter your results to return only `UNTAGGED` images and then pipe that result to a [BatchDeleteImage](#) operation to delete them. Or, you can filter your results to return only `TAGGED` images to list all of the tags in your repository.

## Request Syntax

```
{
  "filter": {
    "tagStatus": "string"
  },
  "maxResults": number,
  "nextToken": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### filter

The filter key and value with which to filter your `ListImages` results.

Type: [ListImagesFilter](#) object

Required: No

### maxResults

The maximum number of image results returned by `ListImages` in paginated output. When this parameter is used, `ListImages` only returns `maxResults` results in a single page along with a `nextToken` response element. The remaining results of the initial request can be seen by sending another `ListImages` request with the returned `nextToken` value. This value can be

between 1 and 1000. If this parameter is not used, then `ListImages` returns up to 100 results and a `nextToken` value, if applicable.

Type: Integer

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

Required: No

### [nextToken](#)

The `nextToken` value returned from a previous paginated `ListImages` request where `maxResults` was used and the results exceeded the value of that parameter. Pagination continues from the end of the previous results that returned the `nextToken` value. This value is `null` when there are no more results to return.

#### **Note**

This token should be treated as an opaque identifier that is only used to retrieve the next items in a list and not for other programmatic purposes.

Type: String

Required: No

### [registryId](#)

The Amazon account ID associated with the registry that contains the repository in which to list images. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

### [repositoryName](#)

The repository with image IDs to be listed.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "imageIds": [
    {
      "imageDigest": "string",
      "imageTag": "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.

### imageIds

The list of image IDs for the requested repository.

Type: Array of [ImageIdentifier](#) objects

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

### nextToken

The nextToken value to include in a future ListImages request. When the results of a ListImages request exceed maxResults, this value can be used to retrieve the next page of results. This value is null when there are no more results to return.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

## InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example lists all of the images in the `amazonlinux` repository.

## Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 33
```

```
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.ListImages
X-Amz-Date: 20161216T200542Z
User-Agent: aws-cli/1.11.22 Python/2.7.12 Darwin/16.3.0 botocore/1.4.79
Content-Type: application/x-amz-json-1.1
Authorization: AWUTHPARAMS

{
  "repositoryName": "amazonlinux"
}
```

## Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 16 Dec 2016 20:05:42 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 235
Connection: keep-alive
x-amzn-RequestId: 05bfc4ac-c3cb-11e6-99fb-b1be070cc24b

{
  "imageIds": [
    {
      "imageDigest":
"sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807",
      "imageTag": "2016.09"
    },
    {
      "imageDigest":
"sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807",
      "imageTag": "latest"
    }
  ]
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)

- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)



# ListTagsForResource

List the tags for an Amazon ECR resource.

## Request Syntax

```
{
  "resourceArn": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### resourceArn

The Amazon Resource Name (ARN) that identifies the resource for which to list the tags. Currently, the only supported resource is an Amazon ECR repository.

Type: String

Required: Yes

## Response Syntax

```
{
  "tags": [
    {
      "Key": "string",
      "Value": "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.

## tags

The tags for the resource.

Type: Array of [Tag](#) objects

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that

you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example lists the tags associated with the `sample-repo` repository.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 81
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.ListTagsForResource
X-Amz-Date: 20161216T201255Z
User-Agent: aws-cli/1.16.310 Python/3.6.1 Darwin/18.7.0 botocore/1.13.46
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "resourceArn": "arn:aws:ecr:us-west-2:012345678910:repository/sample-repo"
}
```

### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 24 Jan 2020 03:48:07 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 11
Connection: keep-alive
x-amzn-RequestId: 3081a92b-2066-41f8-8a47-0580288ada9e

{
  "tags": [
    {
      "Key": "environment",
      "Value": "production"
    }
  ]
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# PutImage

Creates or updates the image manifest and tags associated with an image.

When an image is pushed and all new image layers have been uploaded, the PutImage API is called once to create or update the image manifest and the tags associated with the image.

## Note

This operation is used by the Amazon ECR proxy and is not generally used by customers for pulling and pushing images. In most cases, you should use the docker CLI to pull, tag, and push images.

## Request Syntax

```
{
  "imageDigest": "string",
  "imageManifest": "string",
  "imageManifestMediaType": "string",
  "imageTag": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### imageDigest

The image digest of the image manifest corresponding to the image.

Type: String

Required: No

### imageManifest

The image manifest corresponding to the image to be uploaded.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4194304.

Required: Yes

### imageManifestMediaType

The media type of the image manifest. If you push an image manifest that does not contain the `mediaType` field, you must specify the `imageManifestMediaType` in the request.

Type: String

Required: No

### imageTag

The tag to associate with the image. This parameter is required for images that use the Docker Image Manifest V2 Schema 2 or Open Container Initiative (OCI) formats.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 300.

Required: No

### registryId

The Amazon account ID associated with the registry that contains the repository in which to put the image. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

### repositoryName

The name of the repository in which to put the image.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "image": {
    "imageId": {
      "imageDigest": "string",
      "imageTag": "string"
    },
    "imageManifest": "string",
    "imageManifestMediaType": "string",
    "registryId": "string",
    "repositoryName": "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.

### image

Details of the image uploaded.

Type: [Image](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **ImageAlreadyExistsException**

The specified image has already been pushed, and there were no changes to the manifest or image tag after the last push.

HTTP Status Code: 400

### **ImageDigestDoesNotMatchException**

The specified image digest does not match the digest that Amazon ECR calculated for the image.

HTTP Status Code: 400

### **ImageTagAlreadyExistsException**

The specified image is tagged with a tag that already exists. The repository is configured for tag immutability.

HTTP Status Code: 400

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **KmsException**

The operation failed due to a KMS exception.

HTTP Status Code: 400

### **LayersNotFoundException**

The specified layers could not be found, or the specified layer is not valid for this repository.

HTTP Status Code: 400

### **LimitExceededException**

The operation did not succeed because it would have exceeded a service limit for your account. For more information, see [Amazon ECR service quotas](#) in the Amazon Elastic Container Registry User Guide.

HTTP Status Code: 400

### **ReferencedImagesNotFoundException**

The manifest list is referencing an image that does not exist.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400



## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example puts an image to the `amazonlinux` repository with the tag `2016.09`.

## Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 653
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.PutImage
X-Amz-Date: 20161216T201255Z
User-Agent: aws-cli/1.11.22 Python/2.7.12 Darwin/16.3.0 botocore/1.4.79
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType\n\": \"application/vnd.docker.distribution.manifest.v2+json\n\",,\n  \"config\": {\n    \"mediaType\": \"application/\nvnd.docker.container.image.v1+json\",,\n    \"size\": 1486,\n    \"digest\":\n  \"sha256:5b52b314511a611975c2c65e695d920acdf8ae8848fe0ef00b7d018d1f118b64\"\n  },,\n  \"layers\": [\n    {\n      \"mediaType\": \"application/\nvnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 91768077,\n
```

```

\digest\":
  \sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8\"
  ]\n}\n",
  "repositoryName": "amazonlinux",
  "imageTag": "2016.09"
}

```

## Sample Response

```

HTTP/1.1 200 OK
Server: Server
Date: Fri, 16 Dec 2016 20:12:56 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 786
Connection: keep-alive
x-amzn-RequestId: 084038f1-c3cc-11e6-8d10-9da51cf53fd3

{
  "image": {
    "imageId": {
      "imageDigest":
        "sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807",
      "imageTag": "2016.09"
    },
    "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType
\": \"application/vnd.docker.distribution.manifest.v2+json
\",,\n  \"config\": {\n    \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",,\n    \"size\": 1486,\n    \"digest\":
  \"sha256:5b52b314511a611975c2c65e695d920acdf8ae8848fe0ef00b7d018d1f118b64\"
  },,\n  \"layers\": [\n    {\n      \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",,\n      \"size\": 91768077,\n
  \"digest\":
  \"sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8\"
  ]\n}\n",
    "registryId": "012345678910",
    "repositoryName": "amazonlinux"
  }
}

```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# PutImageScanningConfiguration

## Important

The PutImageScanningConfiguration API is being deprecated, in favor of specifying the image scanning configuration at the registry level. For more information, see [PutRegistryScanningConfiguration](#).

Updates the image scanning configuration for the specified repository.

## Request Syntax

```
{
  "imageScanningConfiguration": {
    "scanOnPush": boolean
  },
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### imageScanningConfiguration

The image scanning configuration for the repository. This setting determines whether images are scanned for known vulnerabilities after being pushed to the repository.

Type: [ImageScanningConfiguration](#) object

Required: Yes

### registryId

The Amazon account ID associated with the registry that contains the repository in which to update the image scanning configuration setting. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository in which to update the image scanning configuration setting.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

## Response Syntax

```
{
  "imageScanningConfiguration": {
    "scanOnPush": boolean
  },
  "registryId": "string",
  "repositoryName": "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.

### imageScanningConfiguration

The image scanning configuration setting for the repository.

Type: [ImageScanningConfiguration](#) object

### registryId

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# PutImageTagMutability

Updates the image tag mutability settings for the specified repository. For more information, see [Image tag mutability](#) in the *Amazon Elastic Container Registry User Guide*.

## Request Syntax

```
{
  "imageTagMutability": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [imageTagMutability](#)

The tag mutability setting for the repository. If MUTABLE is specified, image tags can be overwritten. If IMMUTABLE is specified, all image tags within the repository will be immutable which will prevent them from being overwritten.

Type: String

Valid Values: MUTABLE | IMMUTABLE

Required: Yes

### [registryId](#)

The Amazon account ID associated with the registry that contains the repository in which to update the image tag mutability settings. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No



## repositoryName

The name of the repository in which to update the image tag mutability settings.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "imageTagMutability": "string",
  "registryId": "string",
  "repositoryName": "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.

### imageTagMutability

The image tag mutability setting for the repository.

Type: String

Valid Values: MUTABLE | IMMUTABLE

### registryId

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example updates the image tag mutability setting for the `sample-repo` repository.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 73
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.PutImageTagMutability
X-Amz-Date: 20161216T201255Z
User-Agent: aws-cli/1.16.310 Python/3.6.1 Darwin/18.7.0 botocore/1.13.46
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "repositoryName": "sample-repo",
  "imageTagMutability": "IMMUTABLE"
}
```

### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 24 Jan 2020 03:48:07 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 98
Connection: keep-alive
x-amzn-RequestId: 3081a92b-2066-41f8-8a47-0580288ada9e

{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageTagMutability": "IMMUTABLE"
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# PutLifecyclePolicy

Creates or updates the lifecycle policy for the specified repository. For more information, see [Lifecycle policy template](#).

## Request Syntax

```
{
  "lifecyclePolicyText": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [lifecyclePolicyText](#)

The JSON repository policy text to apply to the repository.

Type: String

Length Constraints: Minimum length of 100. Maximum length of 30720.

Required: Yes

### [registryId](#)

The Amazon account ID associated with the registry that contains the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### [repositoryName](#)

The name of the repository to receive the policy.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "lifecyclePolicyText": "string",
  "registryId": "string",
  "repositoryName": "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.

### [lifecyclePolicyText](#)

The JSON repository policy text.

Type: String

Length Constraints: Minimum length of 100. Maximum length of 30720.

### [registryId](#)

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### [repositoryName](#)

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that

you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

## Example

This example creates a lifecycle policy to expire images older than 14 days for a repository called `project-a/amazon-ecs-sample` in the default registry for an account.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.PutLifecyclePolicy
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.11.144 Python/3.6.1 Darwin/16.6.0 botocore/1.7.2
X-Amz-Date: 20170901T194217Z
Authorization: AUTHPARAMS
Content-Length: 535

{
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\
\n\": 1,\n      \"description\": \"Expire images older than 14 days\",\n
\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n
\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\": \"days\",\n
\n        \"countNumber\": 14\n      },\n      \"action\": {\n
\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}"
```

### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 01 Sep 2017 19:42:18 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 340
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\
\n\": 1,\n      \"description\": \"Expire images older than 14 days\",\n
\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n
\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\": \"days\",\n
\n        \"countNumber\": 14\n      },\n      \"action\": {\n
\n        \"type\": \"expire\"\n      }\n    }\n  ]\n}"
```



```
\ "sinceImagePushed\" , \ "countUnit\" : \ "days\" , \ "countNumber\" : 14 } , \ "action\" : { \ "type\" : \ "expire\" } } } } ,
  \ "registryId\" : \ "012345678910\" ,
  \ "repositoryName\" : \ "project-a/amazon-ecs-sample\"
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# PutRegistryPolicy

Creates or updates the permissions policy for your registry.

A registry policy is used to specify permissions for another Amazon account and is used when configuring cross-account replication. For more information, see [Registry permissions](#) in the *Amazon Elastic Container Registry User Guide*.

## Request Syntax

```
{  
  "policyText": "string"  
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [policyText](#)

The JSON policy text to apply to your registry. The policy text follows the same format as IAM policy text. For more information, see [Registry permissions](#) in the *Amazon Elastic Container Registry User Guide*.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

Required: Yes

## Response Syntax

```
{  
  "policyText": "string",  
  "registryId": "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.

### policyText

The JSON policy text for your registry.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

### registryId

The registry ID.

Type: String

Pattern: [0-9]{12}

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# PutRegistryScanningConfiguration

Creates or updates the scanning configuration for your private registry.

## Request Syntax

```
{
  "rules": [
    {
      "repositoryFilters": [
        {
          "filter": "string",
          "filterType": "string"
        }
      ],
      "scanFrequency": "string"
    }
  ],
  "scanType": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### rules

The scanning rules to use for the registry. A scanning rule is used to determine which repository filters are used and at what frequency scanning will occur.

Type: Array of [RegistryScanningRule](#) objects

Array Members: Minimum number of 0 items. Maximum number of 2 items.

Required: No

### scanType

The scanning type to set for the registry.

When a registry scanning configuration is not defined, by default the BASIC scan type is used. When basic scanning is used, you may specify filters to determine which individual repositories,

or all repositories, are scanned when new images are pushed to those repositories. Alternatively, you can do manual scans of images with basic scanning.

When the ENHANCED scan type is set, Amazon Inspector provides automated vulnerability scanning. You may choose between continuous scanning or scan on push and you may specify filters to determine which individual repositories, or all repositories, are scanned.

Type: String

Valid Values: BASIC | ENHANCED

Required: No

## Response Syntax

```
{
  "registryScanningConfiguration": {
    "rules": [
      {
        "repositoryFilters": [
          {
            "filter": "string",
            "filterType": "string"
          }
        ],
        "scanFrequency": "string"
      }
    ],
    "scanType": "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.

### registryScanningConfiguration

The scanning configuration for your registry.

Type: [RegistryScanningConfiguration](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# PutReplicationConfiguration

Creates or updates the replication configuration for a registry. The existing replication configuration for a repository can be retrieved with the [DescribeRegistry](#) API action. The first time the PutReplicationConfiguration API is called, a service-linked IAM role is created in your account for the replication process. For more information, see [Using service-linked roles for Amazon ECR](#) in the *Amazon Elastic Container Registry User Guide*.

## Note

When configuring cross-account replication, the destination account must grant the source account permission to replicate. This permission is controlled using a registry permissions policy. For more information, see [PutRegistryPolicy](#).

## Request Syntax

```
{
  "replicationConfiguration": {
    "rules": [
      {
        "destinations": [
          {
            "region": "string",
            "registryId": "string"
          }
        ],
        "repositoryFilters": [
          {
            "filter": "string",
            "filterType": "string"
          }
        ]
      }
    ]
  }
}
```



## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [replicationConfiguration](#)

An object representing the replication configuration for a registry.

Type: [ReplicationConfiguration](#) object

Required: Yes

## Response Syntax

```
{
  "replicationConfiguration": {
    "rules": [
      {
        "destinations": [
          {
            "region": "string",
            "registryId": "string"
          }
        ],
        "repositoryFilters": [
          {
            "filter": "string",
            "filterType": "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.

## [replicationConfiguration](#)

The contents of the replication configuration for the registry.

Type: [ReplicationConfiguration](#) object

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)

- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# SetRepositoryPolicy

Applies a repository policy to the specified repository to control access permissions. For more information, see [Amazon ECR Repository policies](#) in the *Amazon Elastic Container Registry User Guide*.

## Request Syntax

```
{
  "force": boolean,
  "policyText": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [force](#)

If the policy you are attempting to set on a repository policy would prevent you from setting another policy in the future, you must force the [SetRepositoryPolicy](#) operation. This is intended to prevent accidental repository lock outs.

Type: Boolean

Required: No

### [policyText](#)

The JSON repository policy text to apply to the repository. For more information, see [Amazon ECR repository policies](#) in the *Amazon Elastic Container Registry User Guide*.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

Required: Yes

## registryId

The Amazon account ID associated with the registry that contains the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: `[0-9]{12}`

Required: No

## repositoryName

The name of the repository to receive the policy.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "policyText": "string",
  "registryId": "string",
  "repositoryName": "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.

### policyText

The JSON repository policy text applied to the repository.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.

### registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### Example

The following example sets a repository policy on the ubuntu repository that allows all Amazon accounts to pull from it.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 223
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.SetRepositoryPolicy
X-Amz-Date: 20151214T235302Z
User-Agent: aws-cli/1.9.10 Python/2.7.10 Darwin/14.5.0 botocore/1.3.10
Content-Type: application/x-amz-json-1.1
Authorization: AWUTHPARAMS

{
  "policyText": "{\"Version\":\"2012-10-17\", \"Statement\": [{\"Sid\": \"AllowPull\",
  \"Effect\": \"Allow\", \"Action\": [\"ecr:BatchGetImage\", \"ecr:GetDownloadUrlForLayer\"],
  \"Principal\": \"*\"}]}\n",
  "repositoryName": "ubuntu"
}
```

### Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Mon, 14 Dec 2015 23:53:02 GMT
```

```
Content-Type: application/x-amz-json-1.1
```

```
Content-Length: 301
```

```
Connection: keep-alive
```

```
x-amzn-RequestId: cfc3ead9-a2bd-11e5-91c7-7126cb670c2b
```

```
{
  "policyText": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement\" : [ {\n    \"Sid\n\" : \"AllowPull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : \"*\",\n    \"Action\" : [ \"ecr:BatchGetImage\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}",
  "registryId": "012345678910",
  "repositoryName": "ubuntu"
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)



# StartImageScan

Starts an image vulnerability scan. An image scan can only be started once per 24 hours on an individual image. This limit includes if an image was scanned on initial push. For more information, see [Image scanning](#) in the *Amazon Elastic Container Registry User Guide*.

## Request Syntax

```
{
  "imageId": {
    "imageDigest": "string",
    "imageTag": "string"
  },
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [imageId](#)

An object with identifying information for an image in an Amazon ECR repository.

Type: [ImageIdentifier](#) object

Required: Yes

### [registryId](#)

The Amazon account ID associated with the registry that contains the repository in which to start an image scan request. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

## repositoryName

The name of the repository that contains the images to scan.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "imageId": {
    "imageDigest": "string",
    "imageTag": "string"
  },
  "imageScanStatus": {
    "description": "string",
    "status": "string"
  },
  "registryId": "string",
  "repositoryName": "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.

### imageId

An object with identifying information for an image in an Amazon ECR repository.

Type: [ImageIdentifier](#) object

### imageScanStatus

The current state of the scan.

Type: [ImageScanStatus](#) object

## registryId

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

## repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **ImageNotFoundException**

The image requested does not exist in the specified repository.

HTTP Status Code: 400

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **LimitExceededException**

The operation did not succeed because it would have exceeded a service limit for your account. For more information, see [Amazon ECR service quotas](#) in the Amazon Elastic Container Registry User Guide.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **UnsupportedImageTypeException**

The image is of a type that cannot be scanned.

HTTP Status Code: 400

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## **Examples**

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### **Example**

This example starts an image scan for and specified by the image digest in the sample-repo repository.

### **Sample Request**

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 141
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.StartImageScan
```

```
X-Amz-Date: 20161216T201255Z
User-Agent: aws-cli/1.16.310 Python/3.6.1 Darwin/18.7.0 botocore/1.13.46
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{
  "repositoryName": "sample-repo",
  "imageId": {
    "imageDigest":
      "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
  }
}
```

## Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 24 Jan 2020 03:48:07 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 208
Connection: keep-alive
x-amzn-RequestId: 3081a92b-2066-41f8-8a47-0580288ada9e

{
  "registryId": "012345678910",
  "repositoryName": "sample-repo",
  "imageId": {
    "imageDigest":
      "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
  },
  "imageScanStatus": {
    "status": "IN_PROGRESS"
  }
}
```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)

- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# StartLifecyclePolicyPreview

Starts a preview of a lifecycle policy for the specified repository. This allows you to see the results before associating the lifecycle policy with the repository.

## Request Syntax

```
{
  "lifecyclePolicyText": "string",
  "registryId": "string",
  "repositoryName": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [lifecyclePolicyText](#)

The policy to be evaluated against. If you do not specify a policy, the current policy for the repository is used.

Type: String

Length Constraints: Minimum length of 100. Maximum length of 30720.

Required: No

### [registryId](#)

The Amazon account ID associated with the registry that contains the repository. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### [repositoryName](#)

The name of the repository to be evaluated.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## Response Syntax

```
{
  "lifecyclePolicyText": "string",
  "registryId": "string",
  "repositoryName": "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.

### lifecyclePolicyText

The JSON repository policy text.

Type: String

Length Constraints: Minimum length of 100. Maximum length of 30720.

### registryId

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### repositoryName

The repository name associated with the request.

Type: String



Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

## status

The status of the lifecycle policy preview request.

Type: String

Valid Values: IN\_PROGRESS | COMPLETE | EXPIRED | FAILED

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **LifecyclePolicyNotFoundException**

The lifecycle policy could not be found, and no policy is set to the repository.

HTTP Status Code: 400

### **LifecyclePolicyPreviewInProgressException**

The previous lifecycle policy preview request has not completed. Wait and try again.

HTTP Status Code: 400

### **RepositoryNotFoundException**

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## Examples

In the following example or examples, the Authorization header contents (AUTHPARAMS) must be replaced with an Amazon Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *Amazon General Reference*.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the [Amazon Command Line Interface \(Amazon CLI\)](#) or one of the [Amazon SDKs](#) to make requests to Amazon, these tools automatically sign the requests for you with the access key that you specify when you configure the tools. When you use these tools, you don't need to learn how to sign requests yourself.

### Example

This example creates a lifecycle policy preview to expire images older than 14 days for a repository called `project-a/amazon-ecs-sample` in the default registry for an account.

### Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.StartLifecyclePolicyPreview
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.11.144 Python/3.6.1 Darwin/16.6.0 botocore/1.7.2
X-Amz-Date: 20170901T221604Z
Authorization: AUTHPARAMS
Content-Length: 535

{
  "repositoryName": "project-a/amazon-ecs-sample",
  "lifecyclePolicyText": "{\n  \"rules\": [\n    {\n      \"rulePriority\n\": 1,\n      \"description\": \"Expire images older than 14 days\",\n      \"selection\": {\n        \"tagStatus\": \"untagged\",\n        \"countType\": \"sinceImagePushed\",\n        \"countUnit\": \"days\",
```

```

        \"countNumber\": 14\n
    },\n
    \"action\": {\n
  \"type\": \"expire\"\n
    }\n
  ]\n}\n"
}

```

## Sample Response

```

HTTP/1.1 200 OK
Server: Server
Date: Fri, 01 Sep 2017 22:16:05 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 583
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "lifecyclePolicyText": "{\n
    \"rules\": [\n
      {\n
        \"rulePriority\n
\": 1,\n
        \"description\": \"Expire images older than 14 days\",\n
        \"selection\": {\n
          \"tagStatus\": \"untagged\",\n
          \"countType\": \"sinceImagePushed\",\n
          \"countUnit\": \"days\",\n
          \"countNumber\": 14\n
        },\n
        \"action\": {\n
          \"type\": \"expire\"\n
        }\n
      }\n
    ]\n  },\n
  \"registryId\": \"012345678910\",\n
  \"repositoryName\": \"project-a/amazon-ecs-sample\",\n
  \"status\": \"IN_PROGRESS\"
}

```

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)

- [Amazon SDK for Ruby V3](#)

# TagResource

Adds specified tags to a resource with the specified ARN. Existing tags on a resource are not changed if they are not specified in the request parameters.

## Request Syntax

```
{
  "resourceArn": "string",
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [resourceArn](#)

The Amazon Resource Name (ARN) of the the resource to which to add tags. Currently, the only supported resource is an Amazon ECR repository.

Type: String

Required: Yes

### [tags](#)

The tags to add to the resource. A tag is an array of key-value pairs. Tag keys can have a maximum character length of 128 characters, and tag values can have a maximum length of 256 characters.

Type: Array of [Tag](#) objects

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](#).

### InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### InvalidTagParameterException

An invalid parameter has been specified. Tag keys can have a maximum character length of 128 characters, and tag values can have a maximum length of 256 characters.

HTTP Status Code: 400

### RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

### ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### TooManyTagsException

The list of tags on the repository is over the limit. The maximum number of tags that can be applied to a repository is 50.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# UntagResource

Deletes specified tags from a resource.

## Request Syntax

```
{  
  "resourceArn": "string",  
  "tagKeys": [ "string" ]  
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### resourceArn

The Amazon Resource Name (ARN) of the resource from which to remove tags. Currently, the only supported resource is an Amazon ECR repository.

Type: String

Required: Yes

### tagKeys

The keys of the tags to be removed.

Type: Array of strings

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](#).



## InvalidParameterException

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

## InvalidTagParameterException

An invalid parameter has been specified. Tag keys can have a maximum character length of 128 characters, and tag values can have a maximum length of 256 characters.

HTTP Status Code: 400

## RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## TooManyTagsException

The list of tags on the repository is over the limit. The maximum number of tags that can be applied to a repository is 50.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)

- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# UpdatePullThroughCacheRule

Updates an existing pull through cache rule.

## Request Syntax

```
{
  "credentialArn": "string",
  "ecrRepositoryPrefix": "string",
  "registryId": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### credentialArn

The Amazon Resource Name (ARN) of the Amazon Secrets Manager secret that identifies the credentials to authenticate to the upstream registry.

Type: String

Length Constraints: Minimum length of 50. Maximum length of 612.

Pattern: `^arn:aws:secretsmanager:[a-zA-Z0-9-:]+:secret:ecr\-\pullthroughcache\[a-zA-Z0-9\|_+=.@-\]+`

Required: Yes

### ecrRepositoryPrefix

The repository name prefix to use when caching images from the source registry.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: Yes

## registryId

The Amazon account ID associated with the registry associated with the pull through cache rule. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

## Response Syntax

```
{
  "credentialArn": "string",
  "ecrRepositoryPrefix": "string",
  "registryId": "string",
  "updatedAt": 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.

### credentialArn

The Amazon Resource Name (ARN) of the Amazon Secrets Manager secret associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 50. Maximum length of 612.

Pattern: ^arn:aws:secretsmanager:[a-zA-Z0-9-:]+:secret:ecr\-\pullthroughcache\[a-zA-Z0-9\\_+=.@-\]+\$

### ecrRepositoryPrefix

The Amazon ECR repository prefix associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

### registryId

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### updatedAt

The date and time, in JavaScript date format, when the pull through cache rule was updated.

Type: Timestamp

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **PullThroughCacheRuleNotFoundException**

The pull through cache rule was not found. Specify a valid pull through cache rule and try again.

HTTP Status Code: 400

### **SecretNotFoundException**

The ARN of the secret specified in the pull through cache rule was not found. Update the pull through cache rule with a valid secret ARN and try again.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

### **UnableToAccessSecretException**

The secret is unable to be accessed. Verify the resource permissions for the secret and try again.

HTTP Status Code: 400

### **UnableToDecryptSecretValueException**

The secret is accessible but is unable to be decrypted. Verify the resource permissions and try again.

HTTP Status Code: 400

### **ValidationException**

There was an exception validating this request.

HTTP Status Code: 400

## **See Also**

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# UploadLayerPart

Uploads an image layer part to Amazon ECR.

When an image is pushed, each new image layer is uploaded in parts. The maximum size of each image layer part can be 20971520 bytes (or about 20MB). The UploadLayerPart API is called once per each new image layer part.

## Note

This operation is used by the Amazon ECR proxy and is not generally used by customers for pulling and pushing images. In most cases, you should use the docker CLI to pull, tag, and push images.

## Request Syntax

```
{
  "layerPartBlob": blob,
  "partFirstByte": number,
  "partLastByte": number,
  "registryId": "string",
  "repositoryName": "string",
  "uploadId": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### layerPartBlob

The base64-encoded layer part payload.

Type: Base64-encoded binary data object

Length Constraints: Minimum length of 0. Maximum length of 20971520.

Required: Yes

### partFirstByte

The position of the first byte of the layer part within the overall image layer.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

### partLastByte

The position of the last byte of the layer part within the overall image layer.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

### registryId

The Amazon account ID associated with the registry to which you are uploading layer parts. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern: [0-9]{12}

Required: No

### repositoryName

The name of the repository to which you are uploading layer parts.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: Yes

### uploadId

The upload ID from a previous [InitiateLayerUpload](#) operation to associate with the layer part upload.



Type: String

Pattern: `[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}`

Required: Yes

## Response Syntax

```
{
  "lastByteReceived": number,
  "registryId": "string",
  "repositoryName": "string",
  "uploadId": "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.

### lastByteReceived

The integer value of the last byte received in the request.

Type: Long

Valid Range: Minimum value of 0.

### registryId

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### repositoryName

The repository name associated with the request.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

## uploadId

The upload ID associated with the request.

Type: String

Pattern: `[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}`

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidLayerPartException**

The layer part size is not valid, or the first byte specified is not consecutive to the last byte of a previous layer part upload.

HTTP Status Code: 400

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **KmsException**

The operation failed due to a KMS exception.

HTTP Status Code: 400

### **LimitExceededException**

The operation did not succeed because it would have exceeded a service limit for your account. For more information, see [Amazon ECR service quotas](#) in the Amazon Elastic Container Registry User Guide.

HTTP Status Code: 400

## RepositoryNotFoundException

The specified repository could not be found. Check the spelling of the specified repository and ensure that you are performing operations on the correct registry.

HTTP Status Code: 400

## ServerException

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## UploadNotFoundException

The upload could not be found, or the specified upload ID is not valid for this repository.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# ValidatePullThroughCacheRule

Validates an existing pull through cache rule for an upstream registry that requires authentication. This will retrieve the contents of the Amazon Secrets Manager secret, verify the syntax, and then validate that authentication to the upstream registry is successful.

## Request Syntax

```
{
  "ecrRepositoryPrefix": "string",
  "registryId": "string"
}
```

## Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#).

The request accepts the following data in JSON format.

### [ecrRepositoryPrefix](#)

The repository name prefix associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern:  $(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*$

Required: Yes

### [registryId](#)

The registry ID associated with the pull through cache rule. If you do not specify a registry, the default registry is assumed.

Type: String

Pattern:  $[0-9]{12}$

Required: No

## Response Syntax

```
{
  "credentialArn": "string",
  "ecrRepositoryPrefix": "string",
  "failure": "string",
  "isValid": boolean,
  "registryId": "string",
  "upstreamRegistryUrl": "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.

### credentialArn

The Amazon Resource Name (ARN) of the Amazon Secrets Manager secret associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 50. Maximum length of 612.

Pattern: `^arn:aws:secretsmanager:[a-zA-Z0-9-:]+:secret:ecr\-\pullthroughcache\[a-zA-Z0-9\/_+=.@-\]+`

### ecrRepositoryPrefix

The Amazon ECR repository prefix associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

### failure

The reason the validation failed. For more details about possible causes and how to address them, see [Using pull through cache rules](#) in the *Amazon Elastic Container Registry User Guide*.

Type: String

### isValid

Whether or not the pull through cache rule was validated. If `true`, Amazon ECR was able to reach the upstream registry and authentication was successful. If `false`, there was an issue and validation failed. The `failureReason` indicates the cause.

Type: Boolean

### registryId

The registry ID associated with the request.

Type: String

Pattern: `[0-9]{12}`

### upstreamRegistryUrl

The upstream registry URL associated with the pull through cache rule.

Type: String

## Errors

For information about the errors that are common to all actions, see [Common Errors](#).

### **InvalidParameterException**

The specified parameter is invalid. Review the available parameters for the API request.

HTTP Status Code: 400

### **PullThroughCacheRuleNotFoundException**

The pull through cache rule was not found. Specify a valid pull through cache rule and try again.

HTTP Status Code: 400

### **ServerException**

These errors are usually caused by a server-side issue.

HTTP Status Code: 500

## ValidationException

There was an exception validating this request.

HTTP Status Code: 400

## See Also

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

- [Amazon Command Line Interface](#)
- [Amazon SDK for .NET](#)
- [Amazon SDK for C++](#)
- [Amazon SDK for Go v2](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for JavaScript V3](#)
- [Amazon SDK for PHP V3](#)
- [Amazon SDK for Python](#)
- [Amazon SDK for Ruby V3](#)

# Data Types

The Amazon Elastic Container Registry API contains several data types that various actions use. This section describes each data type in detail.

## Note

The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- [Attribute](#)
- [AuthorizationData](#)
- [AwsEcrContainerImageDetails](#)
- [CvssScore](#)
- [CvssScoreAdjustment](#)
- [CvssScoreDetails](#)
- [DescribeImagesFilter](#)
- [EncryptionConfiguration](#)
- [EnhancedImageScanFinding](#)
- [Image](#)
- [ImageDetail](#)
- [ImageFailure](#)
- [ImageIdentifier](#)
- [ImageReplicationStatus](#)
- [ImageScanFinding](#)
- [ImageScanFindings](#)
- [ImageScanFindingsSummary](#)
- [ImageScanningConfiguration](#)
- [ImageScanStatus](#)
- [Layer](#)



- [LayerFailure](#)
- [LifecyclePolicyPreviewFilter](#)
- [LifecyclePolicyPreviewResult](#)
- [LifecyclePolicyPreviewSummary](#)
- [LifecyclePolicyRuleAction](#)
- [ListImagesFilter](#)
- [PackageVulnerabilityDetails](#)
- [PullThroughCacheRule](#)
- [Recommendation](#)
- [RegistryScanningConfiguration](#)
- [RegistryScanningRule](#)
- [Remediation](#)
- [ReplicationConfiguration](#)
- [ReplicationDestination](#)
- [ReplicationRule](#)
- [Repository](#)
- [RepositoryFilter](#)
- [RepositoryScanningConfiguration](#)
- [RepositoryScanningConfigurationFailure](#)
- [Resource](#)
- [ResourceDetails](#)
- [ScanningRepositoryFilter](#)
- [ScoreDetails](#)
- [Tag](#)
- [VulnerablePackage](#)

# Attribute

This data type is used in the [ImageScanFinding](#) data type.

## Contents

### key

The attribute key.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

### value

The value assigned to the attribute key.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# AuthorizationData

An object representing authorization data for an Amazon ECR registry.

## Contents

### authorizationToken

A base64-encoded string that contains authorization data for the specified Amazon ECR registry. When the string is decoded, it is presented in the format `user:password` for private registry authentication using `docker login`.

Type: String

Pattern: `^\S+$`

Required: No

### expiresAt

The Unix time in seconds and milliseconds when the authorization token expires. Authorization tokens are valid for 12 hours.

Type: Timestamp

Required: No

### proxyEndpoint

The registry URL to use for this authorization token in a `docker login` command. The Amazon ECR registry URL format is `https://aws_account_id.dkr.ecr.region.amazonaws.com`. For example, `https://012345678910.dkr.ecr.us-east-1.amazonaws.com..`

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# AwsEcrContainerImageDetails

The image details of the Amazon ECR container image.

## Contents

### **architecture**

The architecture of the Amazon ECR container image.

Type: String

Required: No

### **author**

The image author of the Amazon ECR container image.

Type: String

Required: No

### **imageHash**

The image hash of the Amazon ECR container image.

Type: String

Required: No

### **imageTags**

The image tags attached to the Amazon ECR container image.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 300.

Required: No

### **platform**

The platform of the Amazon ECR container image.

Type: String

Required: No

### **pushedAt**

The date and time the Amazon ECR container image was pushed.

Type: Timestamp

Required: No

### **registry**

The registry the Amazon ECR container image belongs to.

Type: String

Pattern: [0-9]{12}

Required: No

### **repositoryName**

The name of the repository the Amazon ECR container image resides in.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: (?:[a-z0-9]+(?:[.\_-][a-z0-9]+)\*/)\*[a-z0-9]+(?:[.\_-][a-z0-9]+)\*

Required: No

## **See Also**

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# CvssScore

The CVSS score for a finding.

## Contents

### baseScore

The base CVSS score used for the finding.

Type: Double

Required: No

### scoringVector

The vector string of the CVSS score.

Type: String

Required: No

### source

The source of the CVSS score.

Type: String

Required: No

### version

The version of CVSS used for the score.

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)

- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# CvssScoreAdjustment

Details on adjustments Amazon Inspector made to the CVSS score for a finding.

## Contents

### **metric**

The metric used to adjust the CVSS score.

Type: String

Required: No

### **reason**

The reason the CVSS score has been adjustment.

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# CvssScoreDetails

Information about the CVSS score.

## Contents

### adjustments

An object that contains details about adjustment Amazon Inspector made to the CVSS score.

Type: Array of [CvssScoreAdjustment](#) objects

Required: No

### score

The CVSS score.

Type: Double

Required: No

### scoreSource

The source for the CVSS score.

Type: String

Required: No

### scoringVector

The vector for the CVSS score.

Type: String

Required: No

### version

The CVSS version used in scoring.

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# DescribeImagesFilter

An object representing a filter on a [DescribeImages](#) operation.

## Contents

### tagStatus

The tag status with which to filter your [DescribeImages](#) results. You can filter results based on whether they are TAGGED or UNTAGGED.

Type: String

Valid Values: TAGGED | UNTAGGED | ANY

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# EncryptionConfiguration

The encryption configuration for the repository. This determines how the contents of your repository are encrypted at rest.

By default, when no encryption configuration is set or the AES256 encryption type is used, Amazon ECR uses server-side encryption with Amazon S3-managed encryption keys which encrypts your data at rest using an AES-256 encryption algorithm. This does not require any action on your part.

For more control over the encryption of the contents of your repository, you can use server-side encryption with Amazon Key Management Service key stored in Amazon Key Management Service (Amazon KMS) to encrypt your images. For more information, see [Amazon ECR encryption at rest](#) in the *Amazon Elastic Container Registry User Guide*.

## Contents

### encryptionType

The encryption type to use.

If you use the KMS encryption type, the contents of the repository will be encrypted using server-side encryption with Amazon Key Management Service key stored in Amazon KMS. When you use Amazon KMS to encrypt your data, you can either use the default Amazon managed Amazon KMS key for Amazon ECR, or specify your own Amazon KMS key, which you already created. For more information, see [Protecting data using server-side encryption with an Amazon KMS key stored in Amazon Key Management Service \(SSE-KMS\)](#) in the *Amazon Simple Storage Service Console Developer Guide*.

If you use the AES256 encryption type, Amazon ECR uses server-side encryption with Amazon S3-managed encryption keys which encrypts the images in the repository using an AES-256 encryption algorithm. For more information, see [Protecting data using server-side encryption with Amazon S3-managed encryption keys \(SSE-S3\)](#) in the *Amazon Simple Storage Service Console Developer Guide*.

Type: String

Valid Values: AES256 | KMS

Required: Yes

## kmsKey

If you use the KMS encryption type, specify the Amazon KMS key to use for encryption. The alias, key ID, or full ARN of the Amazon KMS key can be specified. The key must exist in the same Region as the repository. If no key is specified, the default Amazon managed Amazon KMS key for Amazon ECR will be used.

Type: String

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

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# EnhancedImageScanFinding

The details of an enhanced image scan. This is returned when enhanced scanning is enabled for your private registry.

## Contents

### **awsAccountId**

The Amazon account ID associated with the image.

Type: String

Pattern: [0-9]{12}

Required: No

### **description**

The description of the finding.

Type: String

Required: No

### **findingArn**

The Amazon Resource Number (ARN) of the finding.

Type: String

Required: No

### **firstObservedAt**

The date and time that the finding was first observed.

Type: Timestamp

Required: No

### **lastObservedAt**

The date and time that the finding was last observed.

Type: Timestamp

Required: No

### **packageVulnerabilityDetails**

An object that contains the details of a package vulnerability finding.

Type: [PackageVulnerabilityDetails](#) object

Required: No

### **remediation**

An object that contains the details about how to remediate a finding.

Type: [Remediation](#) object

Required: No

### **resources**

Contains information on the resources involved in a finding.

Type: Array of [Resource](#) objects

Required: No

### **score**

The Amazon Inspector score given to the finding.

Type: Double

Required: No

### **scoreDetails**

An object that contains details of the Amazon Inspector score.

Type: [ScoreDetails](#) object

Required: No

### **severity**

The severity of the finding.

Type: String



Required: No

**status**

The status of the finding.

Type: String

Required: No

**title**

The title of the finding.

Type: String

Required: No

**type**

The type of the finding.

Type: String

Required: No

**updatedAt**

The date and time the finding was last updated at.

Type: Timestamp

Required: No

**See Also**

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# Image

An object representing an Amazon ECR image.

## Contents

### **imageId**

An object containing the image tag and image digest associated with an image.

Type: [ImageIdentifier](#) object

Required: No

### **imageManifest**

The image manifest associated with the image.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 4194304.

Required: No

### **imageManifestMediaType**

The manifest media type of the image.

Type: String

Required: No

### **registryId**

The Amazon account ID associated with the registry containing the image.

Type: String

Pattern: `[0-9]{12}`

Required: No

### **repositoryName**

The name of the repository associated with the image.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ImageDetail

An object that describes an image returned by a [DescribeImages](#) operation.

## Contents

### **artifactMediaType**

The artifact media type of the image.

Type: String

Required: No

### **imageDigest**

The sha256 digest of the image manifest.

Type: String

Required: No

### **imageManifestMediaType**

The media type of the image manifest.

Type: String

Required: No

### **imagePushedAt**

The date and time, expressed in standard JavaScript date format, at which the current image was pushed to the repository.

Type: Timestamp

Required: No

### **imageScanFindingsSummary**

A summary of the last completed image scan.

Type: [ImageScanFindingsSummary](#) object

Required: No

## imageScanStatus

The current state of the scan.

Type: [ImageScanStatus](#) object

Required: No

## imageSizeInBytes

The size, in bytes, of the image in the repository.

If the image is a manifest list, this will be the max size of all manifests in the list.

### Note

Beginning with Docker version 1.9, the Docker client compresses image layers before pushing them to a V2 Docker registry. The output of the `docker images` command shows the uncompressed image size, so it may return a larger image size than the image sizes returned by [DescribeImages](#).

Type: Long

Required: No

## imageTags

The list of tags associated with this image.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 300.

Required: No

## lastRecordedPullTime

The date and time, expressed in standard JavaScript date format, when Amazon ECR recorded the last image pull.

### Note

Amazon ECR refreshes the last image pull timestamp at least once every 24 hours. For example, if you pull an image once a day then the `lastRecordedPullTime` timestamp

will indicate the exact time that the image was last pulled. However, if you pull an image once an hour, because Amazon ECR refreshes the `lastRecordedPullTime` timestamp at least once every 24 hours, the result may not be the exact time that the image was last pulled.

Type: Timestamp

Required: No

### **registryId**

The Amazon account ID associated with the registry to which this image belongs.

Type: String

Pattern: `[0-9]{12}`

Required: No

### **repositoryName**

The name of the repository to which this image belongs.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

## **See Also**

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ImageFailure

An object representing an Amazon ECR image failure.

## Contents

### failureCode

The code associated with the failure.

Type: String

Valid Values: `InvalidImageDigest` | `InvalidImageTag` | `ImageTagDoesNotMatchDigest` | `ImageNotFound` | `MissingDigestAndTag` | `ImageReferencedByManifestList` | `KmsError` | `UpstreamAccessDenied` | `UpstreamTooManyRequests` | `UpstreamUnavailable`

Required: No

### failureReason

The reason for the failure.

Type: String

Required: No

### imageId

The image ID associated with the failure.

Type: [ImageIdentifier](#) object

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)

- [Amazon SDK for Ruby V3](#)



# ImageIdentifier

An object with identifying information for an image in an Amazon ECR repository.

## Contents

### imageDigest

The sha256 digest of the image manifest.

Type: String

Required: No

### imageTag

The tag used for the image.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 300.

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ImageReplicationStatus

The status of the replication process for an image.

## Contents

### failureCode

The failure code for a replication that has failed.

Type: String

Required: No

### region

The destination Region for the image replication.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 25.

Pattern: `[0-9a-z-]{2,25}`

Required: No

### registryId

The Amazon account ID associated with the registry to which the image belongs.

Type: String

Pattern: `[0-9]{12}`

Required: No

### status

The image replication status.

Type: String

Valid Values: `IN_PROGRESS` | `COMPLETE` | `FAILED`

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ImageScanFinding

Contains information about an image scan finding.

## Contents

### attributes

A collection of attributes of the host from which the finding is generated.

Type: Array of [Attribute](#) objects

Array Members: Minimum number of 0 items. Maximum number of 50 items.

Required: No

### description

The description of the finding.

Type: String

Required: No

### name

The name associated with the finding, usually a CVE number.

Type: String

Required: No

### severity

The finding severity.

Type: String

Valid Values: INFORMATIONAL | LOW | MEDIUM | HIGH | CRITICAL | UNDEFINED

Required: No

### uri

A link containing additional details about the security vulnerability.

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ImageScanFindings

The details of an image scan.

## Contents

### **enhancedFindings**

Details about the enhanced scan findings from Amazon Inspector.

Type: Array of [EnhancedImageScanFinding](#) objects

Required: No

### **findings**

The findings from the image scan.

Type: Array of [ImageScanFinding](#) objects

Required: No

### **findingSeverityCounts**

The image vulnerability counts, sorted by severity.

Type: String to integer map

Valid Keys: INFORMATIONAL | LOW | MEDIUM | HIGH | CRITICAL | UNDEFINED

Valid Range: Minimum value of 0.

Required: No

### **imageScanCompletedAt**

The time of the last completed image scan.

Type: Timestamp

Required: No

### **vulnerabilitySourceUpdatedAt**

The time when the vulnerability data was last scanned.

Type: Timestamp

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ImageScanFindingsSummary

A summary of the last completed image scan.

## Contents

### findingSeverityCounts

The image vulnerability counts, sorted by severity.

Type: String to integer map

Valid Keys: INFORMATIONAL | LOW | MEDIUM | HIGH | CRITICAL | UNDEFINED

Valid Range: Minimum value of 0.

Required: No

### imageScanCompletedAt

The time of the last completed image scan.

Type: Timestamp

Required: No

### vulnerabilitySourceUpdatedAt

The time when the vulnerability data was last scanned.

Type: Timestamp

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)





# ImageScanningConfiguration

The image scanning configuration for a repository.

## Contents

### scanOnPush

The setting that determines whether images are scanned after being pushed to a repository. If set to `true`, images will be scanned after being pushed. If this parameter is not specified, it will default to `false` and images will not be scanned unless a scan is manually started with the [API\\_StartImageScan](#) API.

Type: Boolean

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ImageScanStatus

The current status of an image scan.

## Contents

### description

The description of the image scan status.

Type: String

Required: No

### status

The current state of an image scan.

Type: String

Valid Values: IN\_PROGRESS | COMPLETE | FAILED | UNSUPPORTED\_IMAGE | ACTIVE | PENDING | SCAN\_ELIGIBILITY\_EXPIRED | FINDINGS\_UNAVAILABLE

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# Layer

An object representing an Amazon ECR image layer.

## Contents

### **layerAvailability**

The availability status of the image layer.

Type: String

Valid Values: AVAILABLE | UNAVAILABLE

Required: No

### **layerDigest**

The sha256 digest of the image layer.

Type: String

Pattern: `[a-zA-Z0-9-_.]+:[a-fA-F0-9]+`

Required: No

### **layerSize**

The size, in bytes, of the image layer.

Type: Long

Required: No

### **mediaType**

The media type of the layer, such as `application/vnd.docker.image.rootfs.diff.tar.gzip` or `application/vnd.oci.image.layer.v1.tar+gzip`.

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# LayerFailure

An object representing an Amazon ECR image layer failure.

## Contents

### failureCode

The failure code associated with the failure.

Type: String

Valid Values: `InvalidLayerDigest` | `MissingLayerDigest`

Required: No

### failureReason

The reason for the failure.

Type: String

Required: No

### layerDigest

The layer digest associated with the failure.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 1000.

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# LifecyclePolicyPreviewFilter

The filter for the lifecycle policy preview.

## Contents

### tagStatus

The tag status of the image.

Type: String

Valid Values: TAGGED | UNTAGGED | ANY

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# LifecyclePolicyPreviewResult

The result of the lifecycle policy preview.

## Contents

### **action**

The type of action to be taken.

Type: [LifecyclePolicyRuleAction](#) object

Required: No

### **appliedRulePriority**

The priority of the applied rule.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

### **imageDigest**

The sha256 digest of the image manifest.

Type: String

Required: No

### **imagePushedAt**

The date and time, expressed in standard JavaScript date format, at which the current image was pushed to the repository.

Type: Timestamp

Required: No

### **imageTags**

The list of tags associated with this image.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 300.

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# LifecyclePolicyPreviewSummary

The summary of the lifecycle policy preview request.

## Contents

### **expiringImageTotalCount**

The number of expiring images.

Type: Integer

Valid Range: Minimum value of 0.

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# LifecyclePolicyRuleAction

The type of action to be taken.

## Contents

### type

The type of action to be taken.

Type: String

Valid Values: EXPIRE

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ListImagesFilter

An object representing a filter on a [ListImages](#) operation.

## Contents

### tagStatus

The tag status with which to filter your [ListImages](#) results. You can filter results based on whether they are TAGGED or UNTAGGED.

Type: String

Valid Values: TAGGED | UNTAGGED | ANY

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# PackageVulnerabilityDetails

Information about a package vulnerability finding.

## Contents

### cvss

An object that contains details about the CVSS score of a finding.

Type: Array of [CvssScore](#) objects

Required: No

### referenceUrls

One or more URLs that contain details about this vulnerability type.

Type: Array of strings

Required: No

### relatedVulnerabilities

One or more vulnerabilities related to the one identified in this finding.

Type: Array of strings

Required: No

### source

The source of the vulnerability information.

Type: String

Required: No

### sourceUrl

A URL to the source of the vulnerability information.

Type: String

Required: No

**vendorCreatedAt**

The date and time that this vulnerability was first added to the vendor's database.

Type: Timestamp

Required: No

**vendorSeverity**

The severity the vendor has given to this vulnerability type.

Type: String

Required: No

**vendorUpdatedAt**

The date and time the vendor last updated this vulnerability in their database.

Type: Timestamp

Required: No

**vulnerabilityId**

The ID given to this vulnerability.

Type: String

Required: No

**vulnerablePackages**

The packages impacted by this vulnerability.

Type: Array of [VulnerablePackage](#) objects

Required: No

**See Also**

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# PullThroughCacheRule

The details of a pull through cache rule.

## Contents

### **createdAt**

The date and time the pull through cache was created.

Type: Timestamp

Required: No

### **credentialArn**

The ARN of the Secrets Manager secret associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 50. Maximum length of 612.

Pattern: `^arn:aws:secretsmanager:[a-zA-Z0-9-:]+:secret:ecr\-\pullthroughcache\[a-zA-Z0-9\/_+=.@-]+\`

Required: No

### **ecrRepositoryPrefix**

The Amazon ECR repository prefix associated with the pull through cache rule.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 30.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

### **registryId**

The Amazon account ID associated with the registry the pull through cache rule is associated with.

Type: String

Pattern: `[0-9]{12}`

Required: No

### **updatedAt**

The date and time, in JavaScript date format, when the pull through cache rule was last updated.

Type: Timestamp

Required: No

### **upstreamRegistry**

The name of the upstream source registry associated with the pull through cache rule.

Type: String

Valid Values: `ecr-public` | `quay` | `k8s` | `docker-hub` | `github-container-registry` | `azure-container-registry`

Required: No

### **upstreamRegistryUrl**

The upstream registry URL associated with the pull through cache rule.

Type: String

Required: No

## **See Also**

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# Recommendation

Details about the recommended course of action to remediate the finding.

## Contents

### text

The recommended course of action to remediate the finding.

Type: String

Required: No

### url

The URL address to the CVE remediation recommendations.

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# RegistryScanningConfiguration

The scanning configuration for a private registry.

## Contents

### rules

The scanning rules associated with the registry.

Type: Array of [RegistryScanningRule](#) objects

Array Members: Minimum number of 0 items. Maximum number of 2 items.

Required: No

### scanType

The type of scanning configured for the registry.

Type: String

Valid Values: BASIC | ENHANCED

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# RegistryScanningRule

The details of a scanning rule for a private registry.

## Contents

### repositoryFilters

The repository filters associated with the scanning configuration for a private registry.

Type: Array of [ScanningRepositoryFilter](#) objects

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

Required: Yes

### scanFrequency

The frequency that scans are performed at for a private registry. When the ENHANCED scan type is specified, the supported scan frequencies are CONTINUOUS\_SCAN and SCAN\_ON\_PUSH. When the BASIC scan type is specified, the SCAN\_ON\_PUSH scan frequency is supported. If scan on push is not specified, then the MANUAL scan frequency is set by default.

Type: String

Valid Values: SCAN\_ON\_PUSH | CONTINUOUS\_SCAN | MANUAL

Required: Yes

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# Remediation

Information on how to remediate a finding.

## Contents

### recommendation

An object that contains information about the recommended course of action to remediate the finding.

Type: [Recommendation](#) object

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ReplicationConfiguration

The replication configuration for a registry.

## Contents

### rules

An array of objects representing the replication destinations and repository filters for a replication configuration.

Type: Array of [ReplicationRule](#) objects

Array Members: Minimum number of 0 items. Maximum number of 10 items.

Required: Yes

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ReplicationDestination

An array of objects representing the destination for a replication rule.

## Contents

### region

The Region to replicate to.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 25.

Pattern: `[0-9a-z-]{2,25}`

Required: Yes

### registryId

The Amazon account ID of the Amazon ECR private registry to replicate to. When configuring cross-Region replication within your own registry, specify your own account ID.

Type: String

Pattern: `[0-9]{12}`

Required: Yes

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# ReplicationRule

An array of objects representing the replication destinations and repository filters for a replication configuration.

## Contents

### destinations

An array of objects representing the destination for a replication rule.

Type: Array of [ReplicationDestination](#) objects

Array Members: Minimum number of 0 items. Maximum number of 25 items.

Required: Yes

### repositoryFilters

An array of objects representing the filters for a replication rule. Specifying a repository filter for a replication rule provides a method for controlling which repositories in a private registry are replicated.

Type: Array of [RepositoryFilter](#) objects

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

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# Repository

An object representing a repository.

## Contents

### **createdAt**

The date and time, in JavaScript date format, when the repository was created.

Type: Timestamp

Required: No

### **encryptionConfiguration**

The encryption configuration for the repository. This determines how the contents of your repository are encrypted at rest.

Type: [EncryptionConfiguration](#) object

Required: No

### **imageScanningConfiguration**

The image scanning configuration for a repository.

Type: [ImageScanningConfiguration](#) object

Required: No

### **imageTagMutability**

The tag mutability setting for the repository.

Type: String

Valid Values: MUTABLE | IMMUTABLE

Required: No

### **registryId**

The Amazon account ID associated with the registry that contains the repository.

Type: String

Pattern: `[0-9]{12}`

Required: No

### **repositoryArn**

The Amazon Resource Name (ARN) that identifies the repository. The ARN contains the `arn:aws:ecr` namespace, followed by the region of the repository, Amazon account ID of the repository owner, repository namespace, and repository name. For example, `arn:aws:ecr:region:012345678910:repository-namespace/repository-name`.

Type: String

Required: No

### **repositoryName**

The name of the repository.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

### **repositoryUri**

The URI for the repository. You can use this URI for container image push and pull operations.

Type: String

Required: No

## **See Also**

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# RepositoryFilter

The filter settings used with image replication. Specifying a repository filter to a replication rule provides a method for controlling which repositories in a private registry are replicated. If no filters are added, the contents of all repositories are replicated.

## Contents

### filter

The repository filter details. When the PREFIX\_MATCH filter type is specified, this value is required and should be the repository name prefix to configure replication for.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern:  $^(?:[a-z0-9]+(?:[._-][a-z0-9]*)*/)*[a-z0-9]*(?:[._-][a-z0-9]*)*\$$

Required: Yes

### filterType

The repository filter type. The only supported value is PREFIX\_MATCH, which is a repository name prefix specified with the `filter` parameter.

Type: String

Valid Values: PREFIX\_MATCH

Required: Yes

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# RepositoryScanningConfiguration

The details of the scanning configuration for a repository.

## Contents

### **appliedScanFilters**

The scan filters applied to the repository.

Type: Array of [ScanningRepositoryFilter](#) objects

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

Required: No

### **repositoryArn**

The ARN of the repository.

Type: String

Required: No

### **repositoryName**

The name of the repository.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

### **scanFrequency**

The scan frequency for the repository.

Type: String

Valid Values: SCAN\_ON\_PUSH | CONTINUOUS\_SCAN | MANUAL

Required: No

## scanOnPush

Whether or not scan on push is configured for the repository.

Type: Boolean

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# RepositoryScanningConfigurationFailure

The details about any failures associated with the scanning configuration of a repository.

## Contents

### failureCode

The failure code.

Type: String

Valid Values: REPOSITORY\_NOT\_FOUND

Required: No

### failureReason

The reason for the failure.

Type: String

Required: No

### repositoryName

The name of the repository.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 256.

Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*/)*[a-z0-9]+(?:[._-][a-z0-9]+)*`

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)

- [Amazon SDK for Ruby V3](#)

# Resource

Details about the resource involved in a finding.

## Contents

### details

An object that contains details about the resource involved in a finding.

Type: [ResourceDetails](#) object

Required: No

### id

The ID of the resource.

Type: String

Required: No

### tags

The tags attached to the resource.

Type: String to string map

Required: No

### type

The type of resource.

Type: String

Required: No

## See Also

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

- [Amazon SDK for C++](#)

- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ResourceDetails

Contains details about the resource involved in the finding.

## Contents

### awsEcrContainerImage

An object that contains details about the Amazon ECR container image involved in the finding.

Type: [AwsEcrContainerImageDetails](#) object

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ScanningRepositoryFilter

The details of a scanning repository filter. For more information on how to use filters, see [Using filters](#) in the *Amazon Elastic Container Registry User Guide*.

## Contents

### filter

The filter to use when scanning.

Type: String

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

Pattern: `^[a-z0-9*](?:[._\-/a-z0-9*]?[a-z0-9*]+)*$`

Required: Yes

### filterType

The type associated with the filter.

Type: String

Valid Values: WILDCARD

Required: Yes

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# ScoreDetails

Information about the Amazon Inspector score given to a finding.

## Contents

### cvss

An object that contains details about the CVSS score given to a finding.

Type: [CvssScoreDetails](#) object

Required: No

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# Tag

The metadata to apply to a resource to help you categorize and organize them. Each tag consists of a key and a value, both of which you define. Tag keys can have a maximum character length of 128 characters, and tag values can have a maximum length of 256 characters.

## Contents

### Key

One part of a key-value pair that make up a tag. A key is a general label that acts like a category for more specific tag values.

Type: String

Required: Yes

### Value

A value acts as a descriptor within a tag category (key).

Type: String

Required: Yes

## See Also

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)



# VulnerablePackage

Information on the vulnerable package identified by a finding.

## Contents

### **arch**

The architecture of the vulnerable package.

Type: String

Required: No

### **epoch**

The epoch of the vulnerable package.

Type: Integer

Required: No

### **filePath**

The file path of the vulnerable package.

Type: String

Required: No

### **name**

The name of the vulnerable package.

Type: String

Required: No

### **packageManager**

The package manager of the vulnerable package.

Type: String

Required: No

**release**

The release of the vulnerable package.

Type: String

Required: No

**sourceLayerHash**

The source layer hash of the vulnerable package.

Type: String

Required: No

**version**

The version of the vulnerable package.

Type: String

Required: No

**See Also**

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

- [Amazon SDK for C++](#)
- [Amazon SDK for Java V2](#)
- [Amazon SDK for Ruby V3](#)

# 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 [Signing Amazon API requests](#) in the *IAM User Guide*.

## 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 [Create a signed Amazon API request](#) in the *IAM User Guide*.

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 [Elements of an Amazon API request signature](#) in the *IAM User Guide*.

Type: string

Required: Conditional

### **X-Amz-Security-Token**

The temporary security token that was obtained through a call to Amazon Security Token Service (Amazon STS). For a list of services that support temporary security credentials from Amazon STS, see [Amazon Web Services that work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from Amazon STS, 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 [Create a signed Amazon API request](#) in the *IAM User Guide*.

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 Amazon 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 Amazon 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 Amazon access key ID provided does not exist in our records.

HTTP Status Code: 403

## **NotAuthorized**

You do not have permission to perform this action.

HTTP Status Code: 400

## **OptInRequired**

The Amazon 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 Amazon service.

HTTP Status Code: 400