Amazon Elastic Container Registry: API Reference
Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Welcome ........................................................................................................................................... 1
Actions ................................................................................................................................................ 2

BatchCheckLayerAvailability ........................................................................................................ 4
  Request Syntax ................................................................................................................................. 4
  Request Parameters ........................................................................................................................... 4
  Response Syntax ............................................................................................................................... 5
  Response Elements ............................................................................................................................ 5
  Errors .................................................................................................................................................. 5
  Examples ............................................................................................................................................ 6
  See Also ............................................................................................................................................. 7

BatchDeleteImage ............................................................................................................................... 8
  Request Syntax ................................................................................................................................. 8
  Request Parameters ........................................................................................................................... 8
  Response Syntax ............................................................................................................................... 9
  Response Elements ............................................................................................................................ 9
  Errors .................................................................................................................................................. 9
  Examples ............................................................................................................................................ 10
  See Also ............................................................................................................................................. 11

BatchGetImage .................................................................................................................................... 13
  Request Syntax ................................................................................................................................. 13
  Request Parameters ........................................................................................................................... 13
  Response Syntax ............................................................................................................................... 14
  Response Elements ............................................................................................................................ 14
  Errors .................................................................................................................................................. 15
  Examples ............................................................................................................................................ 15
  See Also ............................................................................................................................................. 16

BatchGetRepositoryScanningConfiguration .................................................................................... 17
  Request Syntax ................................................................................................................................. 17
  Request Parameters ........................................................................................................................... 17
  Response Syntax ............................................................................................................................... 17
  Response Elements ............................................................................................................................ 18
  Errors .................................................................................................................................................. 18
  See Also ............................................................................................................................................. 18

CompleteLayerUpload ...................................................................................................................... 20
  Request Syntax ................................................................................................................................. 20
  Request Parameters ........................................................................................................................... 20
  Response Syntax ............................................................................................................................... 21
  Response Elements ............................................................................................................................ 21
  Errors .................................................................................................................................................. 22
  See Also ............................................................................................................................................. 23

CreatePullThroughCacheRule ........................................................................................................... 24
  Request Syntax ................................................................................................................................. 24
  Request Parameters ........................................................................................................................... 24
  Response Syntax ............................................................................................................................... 25
  Response Elements ............................................................................................................................ 25
  Errors .................................................................................................................................................. 25
  See Also ............................................................................................................................................. 26

CreateRepository ............................................................................................................................... 27
  Request Syntax ................................................................................................................................. 27
  Request Parameters ........................................................................................................................... 27
  Response Syntax ............................................................................................................................... 28
  Response Elements ............................................................................................................................ 28
  Errors .................................................................................................................................................. 29
  Examples ............................................................................................................................................ 29
TagResource .................................................................................................................................... 143
Request Syntax ................................................................. 143
Request Parameters ................................................................. 143
Response Elements ................................................................. 143
Errors ................................................................. 143
See Also ................................................................. 144
UntagResource .................................................................................................................................... 145
Request Syntax ................................................................. 145
Request Parameters ................................................................. 145
Response Elements ................................................................. 145
Errors ................................................................. 145
See Also ................................................................. 146
UploadLayerPart .................................................................................................................................... 147
Request Syntax ................................................................. 147
Request Parameters ................................................................. 147
Response Syntax ................................................................. 148
Response Elements ................................................................. 148
Errors ................................................................. 149
See Also ................................................................. 150
Data Types .................................................................................................................................... 151
Attribute .................................................................................................................................... 153
Contents .................................................................................................................................... 153
See Also .................................................................................................................................... 153
AuthorizationData .................................................................................................................................... 154
Contents .................................................................................................................................... 154
See Also .................................................................................................................................... 154
AwsEcrContainerImageDetails .................................................................................................................................... 155
Contents .................................................................................................................................... 155
See Also .................................................................................................................................... 156
CvssScore .................................................................................................................................... 157
Contents .................................................................................................................................... 157
See Also .................................................................................................................................... 157
CvssScoreAdjustment .................................................................................................................................... 158
Contents .................................................................................................................................... 158
See Also .................................................................................................................................... 158
CvssScoreDetails .................................................................................................................................... 159
Contents .................................................................................................................................... 159
See Also .................................................................................................................................... 159
DescribeImagesFilter .................................................................................................................................... 160
Contents .................................................................................................................................... 160
See Also .................................................................................................................................... 160
EncryptionConfiguration .................................................................................................................................... 161
Contents .................................................................................................................................... 161
See Also .................................................................................................................................... 161
EnchancedImageScanFinding .................................................................................................................................... 163
Contents .................................................................................................................................... 163
See Also .................................................................................................................................... 164
Image .................................................................................................................................... 166
Contents .................................................................................................................................... 166
See Also .................................................................................................................................... 166
ImageDetail .................................................................................................................................... 168
Contents .................................................................................................................................... 168
See Also .................................................................................................................................... 169
ImageFailure .................................................................................................................................... 171
Contents .................................................................................................................................... 171
See Also .................................................................................................................................... 171
ImageIdentifier .................................................................................................................................... 172
<table>
<thead>
<tr>
<th>Contents</th>
<th>172</th>
</tr>
</thead>
<tbody>
<tr>
<td>See Also</td>
<td>172</td>
</tr>
<tr>
<td><strong>ImageReplicationStatus</strong></td>
<td>173</td>
</tr>
<tr>
<td>Contents</td>
<td>173</td>
</tr>
<tr>
<td>See Also</td>
<td>173</td>
</tr>
<tr>
<td><strong>ImageScanFinding</strong></td>
<td>174</td>
</tr>
<tr>
<td>Contents</td>
<td>174</td>
</tr>
<tr>
<td>See Also</td>
<td>174</td>
</tr>
<tr>
<td><strong>ImageScanFindings</strong></td>
<td>175</td>
</tr>
<tr>
<td>Contents</td>
<td>175</td>
</tr>
<tr>
<td>See Also</td>
<td>175</td>
</tr>
<tr>
<td><strong>ImageScanFindingsSummary</strong></td>
<td>176</td>
</tr>
<tr>
<td>Contents</td>
<td>176</td>
</tr>
<tr>
<td>See Also</td>
<td>176</td>
</tr>
<tr>
<td><strong>ImageScanningConfiguration</strong></td>
<td>177</td>
</tr>
<tr>
<td>Contents</td>
<td>177</td>
</tr>
<tr>
<td>See Also</td>
<td>177</td>
</tr>
<tr>
<td><strong>ImageScanStatus</strong></td>
<td>178</td>
</tr>
<tr>
<td>Contents</td>
<td>178</td>
</tr>
<tr>
<td>See Also</td>
<td>178</td>
</tr>
<tr>
<td><strong>Layer</strong></td>
<td>179</td>
</tr>
<tr>
<td>Contents</td>
<td>179</td>
</tr>
<tr>
<td>See Also</td>
<td>179</td>
</tr>
<tr>
<td><strong>LayerFailure</strong></td>
<td>180</td>
</tr>
<tr>
<td>Contents</td>
<td>180</td>
</tr>
<tr>
<td>See Also</td>
<td>180</td>
</tr>
<tr>
<td><strong>LifecyclePolicyPreviewFilter</strong></td>
<td>181</td>
</tr>
<tr>
<td>Contents</td>
<td>181</td>
</tr>
<tr>
<td>See Also</td>
<td>181</td>
</tr>
<tr>
<td><strong>LifecyclePolicyPreviewResult</strong></td>
<td>182</td>
</tr>
<tr>
<td>Contents</td>
<td>182</td>
</tr>
<tr>
<td>See Also</td>
<td>182</td>
</tr>
<tr>
<td><strong>LifecyclePolicyPreviewSummary</strong></td>
<td>184</td>
</tr>
<tr>
<td>Contents</td>
<td>184</td>
</tr>
<tr>
<td>See Also</td>
<td>184</td>
</tr>
<tr>
<td><strong>LifecyclePolicyRuleAction</strong></td>
<td>185</td>
</tr>
<tr>
<td>Contents</td>
<td>185</td>
</tr>
<tr>
<td>See Also</td>
<td>185</td>
</tr>
<tr>
<td><strong>ListImagesFilter</strong></td>
<td>186</td>
</tr>
<tr>
<td>Contents</td>
<td>186</td>
</tr>
<tr>
<td>See Also</td>
<td>186</td>
</tr>
<tr>
<td><strong>PackageVulnerabilityDetails</strong></td>
<td>187</td>
</tr>
<tr>
<td>Contents</td>
<td>187</td>
</tr>
<tr>
<td>See Also</td>
<td>188</td>
</tr>
<tr>
<td><strong>PullThroughCacheRule</strong></td>
<td>189</td>
</tr>
<tr>
<td>Contents</td>
<td>189</td>
</tr>
<tr>
<td>See Also</td>
<td>189</td>
</tr>
<tr>
<td><strong>Recommendation</strong></td>
<td>190</td>
</tr>
<tr>
<td>Contents</td>
<td>190</td>
</tr>
<tr>
<td>See Also</td>
<td>190</td>
</tr>
<tr>
<td><strong>RegistryScanningConfiguration</strong></td>
<td>191</td>
</tr>
<tr>
<td>Contents</td>
<td>191</td>
</tr>
<tr>
<td>See Also</td>
<td>191</td>
</tr>
<tr>
<td><strong>RegistryScanningRule</strong></td>
<td>192</td>
</tr>
<tr>
<td>Contents</td>
<td>192</td>
</tr>
<tr>
<td>See Also</td>
<td>192</td>
</tr>
<tr>
<td><strong>Remediation</strong></td>
<td>193</td>
</tr>
</tbody>
</table>
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 AWS General Reference.

This document was last published on May 24, 2022.
Actions

The following actions are supported:

- BatchCheckLayerAvailability (p. 4)
- BatchDeleteImage (p. 8)
- BatchGetImage (p. 13)
- BatchGetRepositoryScanningConfiguration (p. 17)
- CompleteLayerUpload (p. 20)
- CreatePullThroughCacheRule (p. 24)
- CreateRepository (p. 27)
- DeleteLifecyclePolicy (p. 32)
- DeletePullThroughCacheRule (p. 36)
- DeleteRepositoryPolicy (p. 39)
- DeleteRepository (p. 41)
- DeleteRepositoryPolicy (p. 45)
- DescribeImageReplicationStatus (p. 49)
- DescribeImages (p. 52)
- DescribeImageScanFindings (p. 58)
- DescribePullThroughCacheRules (p. 65)
- DescribeRegistry (p. 68)
- DescribeRepositories (p. 70)
- GetAuthorizationToken (p. 74)
- GetDownloadUrlForLayer (p. 77)
- GetLifecyclePolicy (p. 80)
- GetLifecyclePolicyPreview (p. 84)
- GetRepositoryPolicy (p. 89)
- GetRegistryScanningConfiguration (p. 91)
- GetRepositoryPolicy (p. 93)
- InitiateLayerUpload (p. 97)
- ListImages (p. 100)
- ListTagsForResource (p. 104)
- PutImage (p. 107)
- PutImageScanningConfiguration (p. 112)
- PutImageTagMutability (p. 115)
- PutLifecyclePolicy (p. 119)
- PutRegistryPolicy (p. 123)
- PutRegistryScanningConfiguration (p. 125)
- PutReplicationConfiguration (p. 128)
- SetRepositoryPolicy (p. 131)
- StartImageScan (p. 135)
- StartLifecyclePolicyPreview (p. 139)
- TagResource (p. 143)
- UntagResource (p. 145)
• UploadLayerPart (p. 147)
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**

```json
{
    "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** *(p. 4)*
  - 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** *(p. 4)*
  - The AWS 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** *(p. 4)*
  - The name of the repository that is associated with the image layers to check.
  - Type: String
  - Pattern: (?:[a-z0-9]+(?:[-_][a-z0-9]+)*[a-z0-9]+(?:[-_][a-z0-9]+)*)

API Version 2015-09-21
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 (p. 5)
  - Any failures associated with the call.
  - Type: Array of LayerFailure (p. 180) objects
- layers (p. 5)
  - A list of image layer objects corresponding to the image layer references in the request.
  - Type: Array of Layer (p. 179) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

- 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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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 AWS SDKs, see the following:

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

```json
{
    "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 (p. 210).

The request accepts the following data in JSON format.

**imageIds (p. 8)**

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` (p. 172) objects

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

Required: Yes

**registryId (p. 8)**

The AWS 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 (p. 8)**

The repository that contains the image to delete.

Type: String


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

Response Syntax

```json
{
    "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 (p. 9)

Any failures associated with the call.

Type: Array of ImageFailure (p. 171) objects

imageIds (p. 9)

The image IDs of the deleted images.

Type: Array of ImageIdentifier (p. 172) objects

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

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

```plaintext
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

```plaintext
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

```json
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

```json
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 AWS SDKs, see the following:
See Also

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

```json
{
  "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 (p. 210).

The request accepts the following data in JSON format.

acceptedMediaTypes (p. 13)

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 (p. 13)

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 Imageldentifier (p. 172) objects

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

Required: Yes

registryId (p. 13)

The AWS 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 (p. 13)**

The repository that contains the images to describe.

Type: String
Pattern: (?:[a-z0-9]+(?:[._-][a-z0-9]+)+)*[a-z0-9]+(?:[._-][a-z0-9]+)*
Required: Yes

### Response Syntax

```json
{
   "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 (p. 14)**

Any failures associated with the call.
Type: Array of ImageFailure (p. 171) objects

**images (p. 14)**

A list of image objects corresponding to the image references in the request.
Type: Array of Image (p. 166) objects
Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

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

```bash
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": "{"schemaVersion": 2,"mediaType": "application/vnd.docker.distribution.manifest.v2+json","config": {"mediaType": "application/vnd.docker.container.image.v1+json","size": 1486,"digest": "sha256:5b52b314511a611975c2c65e695d920acdf8ae8848fe0ef00b7d018d1f118b64"},"layers": [{"mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip","size": 9176807,"digest": "sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8"},]
            },
            "registryId": "012345678910",
            "repositoryName": "amazonlinux"
        }
    ]
}

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 210).

The request accepts the following data in JSON format.

**repositoryNames (p. 17)**

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.


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
      }
   ]
}
```

API Version 2015-09-21
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 (p. 17)

Any failures associated with the call.

Type: Array of RepositoryScanningConfigurationFailure (p. 202) objects

scanningConfigurations (p. 17)

The scanning configuration for the requested repositories.

Type: Array of RepositoryScanningConfiguration (p. 200) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS SDKs, see the following:

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

Informs 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**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**layerDigests (p. 20)**

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 (p. 20)**

The AWS 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 (p. 20)**

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

Type: String

Response Syntax

```json
{
   "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 (p. 21)**

The sha256 digest of the image layer.

Type: String

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

**registryId (p. 21)**

The registry ID associated with the request.

Type: String

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

**repositoryName (p. 21)**

The repository name associated with the request.

Type: String


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

**uploadId (p. 20)**

The upload ID associated with the layer.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS SDKs, see the following:

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

Creates a pull through cache rule. A pull through cache rule provides a way to cache images from an external public registry in your Amazon ECR private registry.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**ecrRepositoryPrefix (p. 24)**

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

Type: String


Pattern: `[a-zA-Z0-9]+(?:[._-][a-zA-Z0-9]+)*`

Required: Yes

**registryId (p. 24)**

The AWS 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

**upstreamRegistryUrl (p. 24)**

The registry URL of the upstream public registry to use as the source for the pull through cache rule.

Type: String

Required: Yes

Response Syntax

```
{
   "createdAt": number,
   "ecrRepositoryPrefix": "string",
   "registryId": "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 (p. 24)

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

Type: Timestamp
ecrRepositoryPrefix (p. 24)

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

Type: String


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

registryId (p. 24)

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

upstreamRegistryUrl (p. 24)

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 (p. 212).

InvalidParameterException

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

HTTP Status Code: 400

LimitExceeded Exception

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

API Version 2015-09-21

25
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 210).

The request accepts the following data in JSON format.

**encryptionConfiguration (p. 27)**

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

Type: EncryptionConfiguration (p. 161) object

Required: No

**imageScanningConfiguration (p. 27)**

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

Type: ImageScanningConfiguration (p. 177) object

Required: No

**imageTagMutability (p. 27)**

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 (p. 27)**

The AWS 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 (p. 27)**

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).

Type: String


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

Required: Yes

**tags (p. 27)**

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 (p. 207) objects

Required: No

**Response Syntax**

```json
{
    "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 (p. 28)**

- The repository that was created.
- Type: Repository (p. 197) object

## Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

**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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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 AWS SDKs, see the following:

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

Deletes the lifecycle policy associated with the specified repository.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**registryId (p. 32)**

The AWS 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 (p. 32)**

The name of the repository.

Type: String


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

Required: Yes

Response Syntax

```json
{
    "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 (p. 32)**

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

Type: Timestamp

**lifecyclePolicyText (p. 32)**

The JSON lifecycle policy text.

Type: String

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

**registryId (p. 32)**

The registry ID associated with the request.

Type: String

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

**repositoryName (p. 32)**

The repository name associated with the request.

Type: String


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 (p. 212)](#).

**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
Examples

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

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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":[]}"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 AWS SDKs, see the following:

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

Deletes a pull through cache rule.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**ecrRepositoryPrefix (p. 36)**

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

Type: String


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

Required: Yes

**registryId (p. 36)**

The AWS 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

```json
{
   "createdAt": number,
   "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 (p. 36)

The timestamp associated with the pull through cache rule.

Type: Timestamp
ecrRepositoryPrefix (p. 36)

The Amazon ECR repository prefix associated with the request.

Type: String


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

registryId (p. 36)

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

upstreamRegistryUrl (p. 36)

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 (p. 212).

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 AWS SDKs, see the following:

- AWS Command Line Interface
See Also

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteRegistryPolicy

Deletes the registry permissions policy.

Response Syntax

```json
{
   "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 (p. 39)**

The contents of the registry permissions policy that was deleted.

Type: String

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

**registryId (p. 39)**

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 (p. 212).

**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 AWS SDKs, see the following:

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

Deletes a repository. If the repository contains images, you must either delete all images in the repository or use the `force` option to delete the repository.

**Request Syntax**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

- **force (p. 41)**
  
  If a repository contains images, forces the deletion.
  
  Type: Boolean
  
  Required: No

- **registryId (p. 41)**
  
  The AWS 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 (p. 41)**
  
  The name of the repository to delete.
  
  Type: String
  
  
  Pattern: `(?:[a-z0-9]+(?:[._-][a-z0-9]+)*[a-z0-9]+(?:[._-][a-z0-9]+)*)*

  Required: Yes

**Response Syntax**

```
{
  "repository": {
    "createdAt": number,
    ...
  }
}
```

API Version 2015-09-21
"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 (p. 41)**

The repository that was deleted.

Type: Repository (p. 197) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

**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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
See Also

- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
DeleteRepositoryPolicy

Deletes the repository policy associated with the specified repository.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**registryId (p. 45)**

The AWS 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 (p. 45)**

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

Type: String


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

Required: Yes

Response Syntax

```json
{
  "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 (p. 45)

The JSON repository policy that was deleted from the repository.

Type: String

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

registryId (p. 45)

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

repositoryName (p. 45)

The repository name associated with the request.

Type: String


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 (p. 212).

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

ServerErrorException

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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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
{
  "registryId": "012345678910",
  "repositoryName": "ubuntu"
}
```

**See Also**

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

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

Returns the replication status for a specified image.

Request Syntax

```json
{
    "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 (p. 210).

The request accepts the following data in JSON format.

**imageId (p. 49)**

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

Type: ImageIdentifier (p. 172) object

Required: Yes

**registryId (p. 49)**

The AWS 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 (p. 49)**

The name of the repository that the image is in.

Type: String


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

Required: Yes

Response Syntax

```json
{
    "imageId": {
```

API Version 2015-09-21

49
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 (p. 49)**

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

Type: ImageIdentifier (p. 172) object

**replicationStatuses (p. 49)**

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

Type: Array of ImageReplicationStatus (p. 173) objects

**repositoryName (p. 49)**

The repository name associated with the request.

Type: String


**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 212).

**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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 52).

**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 (p. 210).

The request accepts the following data in JSON format.

**filter (p. 52)**

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

Type: DescribeImagesFilter (p. 160) object

Required: No

**imageIds (p. 52)**

The list of image IDs for the requested repository.

Type: Array of ImageIdentifier (p. 172) objects

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

Required: No

**maxResults (p. 52)**

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 (p. 52)

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 (p. 52)

The AWS 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 (p. 52)

The repository that contains the images to describe.

Type: String


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": {
      }
    }
  ]
}
```
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 (p. 53)**

A list of ImageDetail (p. 168) objects that contain data about the image.

Type: Array of ImageDetail (p. 168) objects

**nextToken (p. 53)**

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 (p. 212).

**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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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": "2022-01-24T01:11:45+00:00",
    "vulnerabilitySourceUpdatedAt": "2020-01-23T20:17:00+00:00",
    "findingSeverityCounts": {
        "MEDIUM": 14,
        "INFORMATIONAL": 11,
        "LOW": 27
    }
},
"imageManifestMediaType": "application/vnd.docker.distribution.manifest.v2+json",
"lastRecordedPullTime": "2022-02-23T07:35:16.530000+00:00"
}
]

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
• AWS 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 (p. 210).

The request accepts the following data in JSON format.

**imageId (p. 58)**

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

Type: ImageIdentifier (p. 172) object

Required: Yes

**maxResults (p. 58)**

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 (p. 58)**

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 (p. 58)

The AWS 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 (p. 58)

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

Type: String


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",
              "packageName": "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"
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 (p. 59)**

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

Type: ImageIdentifier (p. 172) object

**imageScanFindings (p. 59)**

The information contained in the image scan findings.

Type: ImageScanFindings (p. 175) object

**imageScanStatus (p. 59)**

The current state of the scan.

Type: ImageScanStatus (p. 178) object

**nextToken (p. 59)**

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 (p. 59)**

The registry ID associated with the request.

Type: String

Pattern: [0–9]{12}

**repositoryName (p. 59)**

The repository name associated with the request.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

**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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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
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.",
        "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": "sha256:74b2c688c700ec95a93e478c6db959737c148df3f65e706abe0318726e885e6"
},
"imageScanStatus": {
    "status": "COMPLETE",
    "description": "The scan was completed successfully."
}

See Also

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

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

Returns the pull through cache rules for a registry.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**ecrRepositoryPrefixes (p. 65)**

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.


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

Required: No

**maxResults (p. 65)**

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 (p. 65)**

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 (p. 65)

The AWS 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

```json
{
  "nextToken": "string",
  "pullThroughCacheRules": [
    {
      "createdAt": number,
      "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.

nextToken (p. 66)

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 (p. 66)

The details of the pull through cache rules.

Type: Array of PullThroughCacheRule (p. 189) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

InvalidParameterValue

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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 128) 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 (p. 68)

The ID of the registry.

Type: String

Pattern: [0–9]{12}

replicationConfiguration (p. 68)

The replication configuration for the registry.

Type: ReplicationConfiguration (p. 194) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS SDKs, see the following:

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

Describes image repositories in a registry.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**maxResults (p. 70)**

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 (p. 70)**

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 (p. 70)**

The AWS 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
Response Syntax

```json
{
    "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 (p. 71)**

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 (p. 71)**

A list of repository objects corresponding to valid repositories.

Type: Array of `Repository` objects (p. 197) objects
Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 AWS CLI offers an `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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

`registryIds (p. 74)`

This parameter has been deprecated.

A list of AWS 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

```json
{
  "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 (p. 74)**

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

Type: Array of [AuthorizationData (p. 154)] objects

### Errors

For information about the errors that are common to all actions, see [Common Errors (p. 212)].

**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 AWS Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process in the AWS General Reference](#).

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

[API Version 2015-09-21]
See Also

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

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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**layerDigest (p. 77)**

The digest of the image layer to download.

Type: String

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

Required: Yes

**registryId (p. 77)**

The AWS 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 (p. 77)**

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

Type: String


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 (p. 78)**

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

Type: String

**layerDigest (p. 78)**

The digest of the image layer to download.

Type: String

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

**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

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 210).

The request accepts the following data in JSON format.

**registryId (p. 80)**

The AWS 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 (p. 80)**

The name of the repository.

Type: String


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 (p. 80)**

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

Type: Timestamp

**lifecyclePolicyText (p. 80)**

The JSON lifecycle policy text.

Type: String

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

**registryId (p. 80)**

The registry ID associated with the request.

Type: String

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

**repositoryName (p. 80)**

The repository name associated with the request.

Type: String


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 (p. 212)](#).

**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
Examples

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

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

```plaintext
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

```plaintext
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 AWS SDKs, see the following:

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

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

Request Syntax

```json
{
    "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 (p. 210).

The request accepts the following data in JSON format.

**filter (p. 84)**

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

Type: LifecyclePolicyPreviewFilter (p. 181) object

Required: No

**imageIds (p. 84)**

The list of imageIds to be included.

Type: Array of ImageIdentifier (p. 172) objects

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

Required: No

**maxResults (p. 84)**

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 (p. 84)`

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 (p. 84)`

The AWS 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 (p. 84)`

The name of the repository.

Type: String


Required: Yes

### Response Syntax

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

API Version 2015-09-21
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 (p. 85)**

The JSON lifecycle policy text.

Type: String

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

**nextToken (p. 85)**

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 (p. 85)**

The results of the lifecycle policy preview request.

Type: Array of LifecyclePolicyPreviewResult (p. 182) objects

**registryId (p. 85)**

The registry ID associated with the request.

Type: String

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

**repositoryName (p. 85)**

The repository name associated with the request.

Type: String


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

**status (p. 85)**

The status of the lifecycle policy preview request.

Type: String

Valid Values: IN_PROGRESS | COMPLETE | EXPIRED | FAILED

**summary (p. 85)**

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

Type: LifecyclePolicyPreviewSummary (p. 184) object
Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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

Examples

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

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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: 20170901T222305Z
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":"{
        "rules": [
            {
                "rulePriority": 1,
                "description": "Expire images older than 14 days",
                "selection": {
                    "tagStatus": "untagged",
                    "countType": "sinceImagePushed",
                    "countUnit": "days",
                    "countNumber": 14
                },
                "action": {
                    "type": "expire"
                }
            }
        
            "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 AWS SDKs, see the following:

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

Retrieves the permissions policy for a registry.

Response Syntax

```json
{
    "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 (p. 89)**

The JSON text of the permissions policy for a registry.

Type: String

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

**registryId (p. 89)**

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 (p. 212).

**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 AWS SDKs, see the following:

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

Retrieves the scanning configuration for a registry.

**Response Syntax**

```json
{
  "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 (p. 91)**
  The ID of the registry.
  - Type: String
  - Pattern: [0–9]{12}

- **scanningConfiguration (p. 91)**
  The scanning configuration for the registry.
  - Type: RegistryScanningConfiguration (p. 191) object

**Errors**

For information about the errors that are common to all actions, see [Common Errors (p. 212)].

- **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 AWS SDKs, see the following:

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

Retrieves the repository policy for the specified repository.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

registryId (p. 93)

The AWS 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 (p. 93)

The name of the repository with the policy to retrieve.

Type: String


Required: Yes

Response Syntax

```json
{
  "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 (p. 93)

The JSON repository policy text associated with the repository.

Type: String

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

registryId (p. 93)

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

repositoryName (p. 93)

The repository name associated with the request.

Type: String


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 (p. 212).

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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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
{
  "registryId": "012345678910",
  "repositoryName": "ubuntu"
}
```

**See Also**

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 210).

The request accepts the following data in JSON format.

**registryId** (p. 97)

The AWS 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** (p. 97)

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

- Type: String
- 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 (p. 97)**

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

Type: Long

Valid Range: Minimum value of 0.

**uploadId (p. 97)**

The upload ID for the layer upload. This parameter is passed to further UploadLayerPart (p. 147) and CompleteLayerUpload (p. 20) 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 (p. 212).

**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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS 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 (p. 8) 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 (p. 210).

The request accepts the following data in JSON format.

**filter (p. 100)**

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

Type: ListImagesFilter (p. 186) object

Required: No

**maxResults (p. 100)**

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 (p. 100)**

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 (p. 100)
The AWS 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 (p. 100)
The repository with image IDs to be listed.

Type: String
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 (p. 101)
The list of image IDs for the requested repository.

Type: Array of ImageIdentifier (p. 172) objects

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

nextToken (p. 101)
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.
Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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
```
See Also

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

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

List the tags for an Amazon ECR resource.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

resourceArn (p. 104)

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

```json
{
  "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 (p. 104)

The tags for the resource.

Type: Array of Tag (p. 207) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).
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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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
```
See Also

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

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

```json
{
  "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 (p. 210).

The request accepts the following data in JSON format.

- **imageDigest (p. 107)**
  
  The image digest of the image manifest corresponding to the image.

  Type: String

  Required: No

- **imageManifest (p. 107)**
  
  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 (p. 107)**
  
  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 (p. 107)**
  
  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 (p. 107)

The AWS 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 (p. 107)

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

Type: String
Pattern: (?:[a-z0-9]+(?:[._-][a-z0-9]*)*)*[a-z0-9]+(?:[._-][a-z0-9]*)*
Required: Yes

Response Syntax

```json
{
   "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 (p. 108)

Details of the image uploaded.
Type: Image (p. 166) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).
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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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": "/application/vnd.docker.distribution.manifest.v2+json"},n "config": {n "mediaType": "/application/vnd.docker.container.image.v1+json"},n "size": 1486,n "digest": "sha256:5b52b31451a611975c2c65e695d920acdf8ae8848fe0e00b7d018d1f118b64"},n "layers": [{n "mediaType": "/application/vnd.docker.image.rootfs.diff.tar.gzip"},n "size": 9176077,n "digest": "sha256:8e3fa21c4cc40232e835a6761332d225c7af335cf55f44ada2ed90e4ab7e8"},n "repositoryName": "amazonlinux",n "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:f1d4ae3f7261a72e98c6ebeff9885cf10a0e5b762585a43e0700ed99863807",
         "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": "sha256:5b52b31451a611975c2c65e695d920acdf8ae8848fe0e00b7d018d1f118b64"},n "layers": [{n "mediaType": "/application/vnd.docker.image.rootfs.diff.tar.gzip"},n "size": 9176077,n "digest": "sha256:8e3fa21c4cc40232e835a6761332d225c7af335cf55f44ada2ed90e4ab7e8"},n "repositoryName": "amazonlinux",n "imageTag": "2016.09"
}
```
"digest": "sha256:5b52b314511a611975c2c65e695d920acdf8ae8848fe0ef00b7d018d1f118b64",

"layers": [ {
   "mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip",
   "size": 91768077,
   "digest": "sha256:8e3fa21c4cc40232e35a6761332d225c7af3325c5755f44ada2ed9d0e4ab7e8"
} ]

"registryId": "012345678910",
"repositoryName": "amazonlinux"

See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 125)`.

Updates the image scanning configuration for the specified repository.

**Request Syntax**

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

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**imageScanningConfiguration (p. 112)**

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 (p. 177)` object

Required: Yes

**registryId (p. 112)**

The AWS 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 (p. 112)**

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

Type: String


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

Required: Yes
Response Syntax

```json
{
   "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 (p. 113)**

The image scanning configuration setting for the repository.

Type: ImageScanningConfiguration (p. 177) object

**registryId (p. 113)**

The registry ID associated with the request.

Type: String

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

**repositoryName (p. 113)**

The repository name associated with the request.

Type: String


Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

**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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 210).

The request accepts the following data in JSON format.

**imageTagMutability (p. 115)**

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 (p. 115)**

The AWS 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 (p. 115)**

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

Type: String


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

Required: Yes

Response Syntax

```
{
}
```
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 (p. 115)**

The image tag mutability setting for the repository.

Type: String

Valid Values: MUTABLE | IMMUTABLE

**registryId (p. 115)**

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

**repositoryName (p. 115)**

The repository name associated with the request.

Type: String


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 (p. 212).

**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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.

You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS 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 (p. 210).

The request accepts the following data in JSON format.

**lifecyclePolicyText (p. 119)**

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 (p. 119)**

The AWS 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 (p. 119)**

The name of the repository to receive the policy.

Type: String


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 (p. 119)**
  - The JSON repository policy text.
  - Type: String
  - Length Constraints: Minimum length of 100. Maximum length of 30720.

- **registryId (p. 119)**
  - The registry ID associated with the request.
  - Type: String
  - Pattern: [0-9]{12}

- **repositoryName (p. 119)**
  - The repository name associated with the request.
  - Type: String
  - 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 (p. 212).

- **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 AWS Signature Version 4 signature. For more information about creating these signatures, see Signature Version 4 Signing Process in the AWS General Reference.
You only need to learn how to sign HTTP requests if you intend to manually create them. When you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs to make requests to AWS, 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": "{
        "rules": [
            {
                "rulePriority": 1,
                "description": "Expire images older than 14 days",
                "selection": {
                    "tagStatus": "untagged",
                    "countType": "sinceImagePushed",
                    "countUnit": "days",
                    "countNumber": 14
                },
                "action": {
                    "type": "expire"
                }
            }
        ]
    }
}
```

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

**See Also**

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
See Also

- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3
PutRegistryPolicy

Creates or updates the permissions policy for your registry.

A registry policy is used to specify permissions for another AWS 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**policyText (p. 123)**

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

```json
{
    "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 (p. 123)**

The JSON policy text for your registry.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 10240.
registryId (p. 123)
   The registry ID.
   Type: String
   Pattern: [0-9]{12}

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS SDKs, see the following:

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

Creates or updates the scanning configuration for your private registry.

Request Syntax

```json
{
  "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 (p. 210).

The request accepts the following data in JSON format.

rules (p. 125)

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 (p. 192) objects

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

Required: No

scanType (p. 125)

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

```json
{
    "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 (p. 126)

- The scanning configuration for your registry.
- Type: RegistryScanningConfiguration (p. 191) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS SDKs, see the following:

- AWS Command Line Interface
• AWS SDK for .NET
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for JavaScript
• AWS SDK for PHP V3
• AWS SDK for Python
• AWS 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 (p. 68) 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 (p. 123).

Request Syntax

```json
{
   "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 (p. 210).

The request accepts the following data in JSON format.

replicationConfiguration (p. 128)

An object representing the replication configuration for a registry.

Type: ReplicationConfiguration (p. 194) object

Required: Yes

Response Syntax

```json
{
   "replicationConfiguration": {
      "rules": [
```

API Version 2015-09-21
128
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 (p. 128)

The contents of the replication configuration for the registry.

Type: ReplicationConfiguration (p. 194) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 212).

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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
See Also

- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**force (p. 131)**

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 (p. 131) operation. This is intended to prevent accidental repository lock outs.

Type: Boolean

Required: No

**policyText (p. 131)**

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 (p. 131)**

The AWS 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 (p. 131)**

The name of the repository to receive the policy.

Type: String

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 (p. 132)**

The JSON repository policy text applied to the repository.

Type: String

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

**registryId (p. 132)**

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

**repositoryName (p. 132)**

The repository name associated with the request.

Type: String


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 (p. 212).

**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 AWS Signature Version 4 signature. For more information about creating these signatures, see [Signature Version 4 Signing Process](#) in the *AWS General Reference*.

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

#### Sample Request

```http
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

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

#### Sample Response

```http
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

{
  "registryId": "012345678910",
  "repositoryName": "ubuntu"
}
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 210).

The request accepts the following data in JSON format.

**imageId (p. 135)**

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

Type: ImageIdentifier (p. 172) object

Required: Yes

**registryId (p. 135)**

The AWS 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 (p. 135)**

The name of the repository that contains the images to scan.

Type: String


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

Required: Yes

Response Syntax

```
{
}
```
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 (p. 135)**

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

Type: ImageIdentifier (p. 172) object

**imageScanStatus (p. 135)**

The current state of the scan.

Type: ImageScanStatus (p. 178) object

**registryId (p. 135)**

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

**repositoryName (p. 135)**

The repository name associated with the request.

Type: String


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 (p. 212).

**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: 500  
**ServerException**

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

HTTP Status Code: 400  
**UnsupportedImageTypeException**

The image is of a type that cannot be scanned.

HTTP Status Code: 400  
**ValidationException**

There was an exception validating this request.

Examples

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

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

```plaintext
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
```
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 AWS SDKs, see the following:

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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

**lifecyclePolicyText (p. 139)**

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 (p. 139)**

The AWS 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 (p. 139)**

The name of the repository to be evaluated.

Type: String


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

Required: Yes

Response Syntax

```json
{
```
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 (p. 139)**

The JSON repository policy text.

Type: String

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

**registryId (p. 139)**

The registry ID associated with the request.

Type: String

Pattern: [0-9]{12}

**repositoryName (p. 139)**

The repository name associated with the request.

Type: String


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

**status (p. 139)**

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 (p. 212).

**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

Examples

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

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

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

Sample Response

```
HTTP/1.1 200 OK
```
See Also

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

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 210).

The request accepts the following data in JSON format.

**resourceArn (p. 143)**

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 (p. 143)**

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 (p. 207) 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 (p. 212).

**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 AWS SDKs, see the following:

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

Deletes specified tags from a resource.

**Request Syntax**

```json
{
    "resourceArn": "string",
    "tagKeys": [ "string" ]
}
```

**Request Parameters**

For information about the parameters that are common to all actions, see Common Parameters (p. 210).

The request accepts the following data in JSON format.

- **resourceArn (p. 145)**
  
  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 (p. 145)**
  
  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 (p. 212).

- **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 AWS SDKs, see the following:

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

```json
{
  "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 (p. 147)**

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 (p. 147)**

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 (p. 147)**

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 (p. 147)**

The AWS 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 (p. 147)**

The name of the repository to which you are uploading layer parts.

Type: String


Required: Yes

**uploadId (p. 147)**

The upload ID from a previous InitiateLayerUpload (p. 97) 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**

```json
{
  "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 (p. 148)**

The integer value of the last byte received in the request.

Type: Long

Valid Range: Minimum value of 0.

**registryId (p. 148)**

The registry ID associated with the request.
Type: String

Pattern: [0-9]{12}

**repositoryName (p. 148)**

The repository name associated with the request.

Type: String


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

**uploadId (p. 148)**

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 (p. 212).

**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 AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS 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 (p. 153)
- AuthorizationData (p. 154)
- AwsEcrContainerImageDetails (p. 155)
- CvssScore (p. 157)
- CvssScoreAdjustment (p. 158)
- CvssScoreDetails (p. 159)
- DescribeImagesFilter (p. 160)
- EncryptionConfiguration (p. 161)
- EnhancedImageScanFinding (p. 163)
- Image (p. 166)
- ImageDetail (p. 168)
- ImageFailure (p. 171)
- ImageIdentifier (p. 172)
- ImageReplicationStatus (p. 173)
- ImageScanFinding (p. 174)
- ImageScanFindings (p. 175)
- ImageScanFindingsSummary (p. 176)
- ImageScanningConfiguration (p. 177)
- ImageScanStatus (p. 178)
- Layer (p. 179)
- LayerFailure (p. 180)
- LifecyclePolicyPreviewFilter (p. 181)
- LifecyclePolicyPreviewResult (p. 182)
- LifecyclePolicyPreviewSummary (p. 184)
- LifecyclePolicyRuleAction (p. 185)
- ListImagesFilter (p. 186)
- PackageVulnerabilityDetails (p. 187)
- PullThroughCacheRule (p. 189)
- Recommendation (p. 190)
- RegistryScanningConfiguration (p. 191)
- RegistryScanningRule (p. 192)
- Remediation (p. 193)
- ReplicationConfiguration (p. 194)
- ReplicationDestination (p. 195)
- ReplicationRule (p. 196)
• Repository (p. 197)
• RepositoryFilter (p. 199)
• RepositoryScanningConfiguration (p. 200)
• RepositoryScanningConfigurationFailure (p. 202)
• Resource (p. 203)
• ResourceDetails (p. 204)
• ScanningRepositoryFilter (p. 205)
• ScoreDetails (p. 206)
• Tag (p. 207)
• VulnerablePackage (p. 208)
Attribute

This data type is used in the ImageScanFinding (p. 174) data type.

Contents

key

The attribute key.

Type: String


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 AWS SDKs, see the following:

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


Type: String
Required: No

See Also

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

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


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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 158) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
DescribeImagesFilter

An object representing a filter on a DescribeImages (p. 52) operation.

Contents

tagStatus

The tag status with which to filter your DescribeImages (p. 52) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS Key Management Service key stored in AWS Key Management Service (AWS 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 AWS Key Management Service key stored in AWS KMS. When you use AWS KMS to encrypt your data, you can either use the default AWS managed AWS KMS key for Amazon ECR, or specify your own AWS KMS key, which you already created. For more information, see Protecting data using server-side encryption with an AWS KMS key stored in AWS 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 AWS KMS key to use for encryption. The alias, key ID, or full ARN of the AWS KMS key can be specified. The key must exist in the same Region as the repository. If no key is specified, the default AWS managed AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
See Also

- AWS SDK for Java V2
- AWS 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 AWS 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 (p. 187) object
Required: No

remediation

An object that contains the details about how to remediate a finding.
Type: Remediation (p. 193) object
Required: No
resources
Contains information on the resources involved in a finding.
Type: Array of Resource (p. 203) 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 (p. 206) 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 AWS SDKs, see the following:
• AWS SDK for C++
• AWS SDK for Go
• AWS SDK for Java V2
• AWS SDK for Ruby V3
An object representing an Amazon ECR image.

Contents

imageId

An object containing the image tag and image digest associated with an image.

Type: ImageIdentifier (p. 172) 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 AWS 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


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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
See Also

- AWS SDK for Java V2
- AWS SDK for Ruby V3
ImageDetail

An object that describes an image returned by a DescribeImages (p. 52) 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 (p. 176) object

Required: No

imageScanStatus

The current state of the scan.

Type: ImageScanStatus (p. 178) 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* (p. 52).

**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 AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
• AWS 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

Required: No

failureReason

The reason for the failure.

Type: String

Required: No

imageId

The image ID associated with the failure.

Type: ImageIdentifier (p. 172) object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

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


Pattern: [0-9a-z-]{2,25}

Required: No

registryId

The AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 153) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 163) objects

Required: No

findings

The findings from the image scan.

Type: Array of ImageScanFinding (p. 174) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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.gz 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
LifecyclePolicyPreviewResult

The result of the lifecycle policy preview.

Contents

action

The type of action to be taken.
Type: LifecyclePolicyRuleAction (p. 185) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3
ListImagesFilter

An object representing a filter on a ListImages (p. 100) operation.

Contents

tagStatus

The tag status with which to filter your ListImages (p. 100) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 157) 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

cvss
An object that contains details about the CVSS score of a finding.

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 (p. 208) objects
Required: No

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

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

ecrRepositoryPrefix

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

Type: String
Pattern: [a-z0-9]+(?:[._-][a-z0-9]+)*
Required: No

registryId

The AWS account ID associated with the registry the pull through cache rule is associated with.

Type: String
Pattern: [0-9]{12}
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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 192) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 205) 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 and MANUAL scan frequencies are supported.

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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 190) object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 196) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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
Pattern: [0-9a-z-]{2,25}
Required: Yes

registryId

The AWS 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 195) 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 (p. 199) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 161) object

Required: No

imageScanningConfiguration

The image scanning configuration for a repository.

Type: ImageScanningConfiguration (p. 177) object

Required: No

imageTagMutability

The tag mutability setting for the repository.

Type: String

Valid Values: MUTABLE | IMMUTABLE

Required: No

registryId

The AWS 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, AWS account ID of the repository owner, repository namespace, and repository name. For example, arn:aws:ecr:region:012345678910:repository/test.

Type: String

Required: No

repositoryName

The name of the repository.
Type: String


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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 repository filter is specified, all images in the repository 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


Pattern: `^([^/]+(/[._-][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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 205) 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


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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
• AWS SDK for Java V2
• AWS 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


Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 204) 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 155) object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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
  - 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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 (p. 159) object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 AWS SDKs, see the following:

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

dsourceLayerHash
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 AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS 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 Signature Version 4 Signing Process in the Amazon Web Services General Reference.

**Action**

The action to be performed.

Type: string

Required: Yes

**Version**

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

Type: string

Required: Yes

**X-Amz-Algorithm**

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

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

Type: string

Valid Values: AWS4-HMAC-SHA256

Required: Conditional

**X-Amz-Credential**

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

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

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

Type: string

Required: Conditional

**X-Amz-Date**

The date that is used to create the signature. The format must be ISO 8601 basic format (’YYYYMMDD’/’THHMMSS’Z’). For example, the following date time is a valid X-Amz-Date value: 20120325T120000Z.

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

Type: string  
Required: Conditional

X-Amz-Security-Token

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

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

Type: string  
Required: Conditional

X-Amz-Signature

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

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

Type: string  
Required: Conditional

X-Amz-SignedHeaders

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

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

Type: string  
Required: Conditional
Common Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

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

HTTP Status Code: 500

InvalidAction

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

HTTP Status Code: 400

InvalidClientTokenId

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

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

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

HTTP Status Code: 400

InvalidQueryParameter

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

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400
MissingAuthenticationToken

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

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

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

HTTP Status Code: 403

RequestExpired

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

HTTP Status Code: 400

ServiceUnavailable

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

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationException

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

HTTP Status Code: 400