Amazon Elastic Container Registry

API Reference

API Version 2015-09-21

Amazon Elastic Container Registry: API Reference

Table of Contents

Welcome

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

This document was last published on June 3, 2020.

Actions

The following actions are supported:

BatchCheckLayerAvailability

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

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

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

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

layerDigests (p. 3)

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. 3)

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. 3)

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

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

failures (p. 4)

Any failures associated with the call.

Type: Array of LayerFailure (p. 132) objects

layers (p. 4)

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

Type: Array of Layer (p. 131) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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: 20161216T195733Z
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
- 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

```
{
   "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. 142).

The request accepts the following data in JSON format.

imageIds (p. 7)

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. 125) objects

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

Required: Yes

registryId (p. 7)

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. 7)

The repository that contains the image to delete.

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

failures (p. 8)

Any failures associated with the call.

Type: Array of ImageFailure (p. 124) objects

imageIds (p. 8)

The image IDs of the deleted images.

Type: Array of ImageIdentifier (p. 125) 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. 144).

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

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

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 16 Dec 2016 19:37:11 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 138
Connection: keep-alive
x-amzn-RequestId: 09cc7023-c3c7-11e6-8acf-61b7dd8abe56

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

```
    }
  ]
}
```

Example

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

Sample Request

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

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

Sample Response

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

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

See Also

For more information about using this API in one of the language-specific 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
- 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

```
{
   "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. 142).

The request accepts the following data in JSON format.

acceptedMediaTypes (p. 12)

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. 12)

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

Type: Array of ImageIdentifier (p. 125) objects

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

Required: Yes

registryId (p. 12)

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. 12)

The repository that contains the images to describe.

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

failures (p. 13)

Any failures associated with the call.

Type: Array of ImageFailure (p. 124) objects

images (p. 13)

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

Type: Array of Image (p. 120) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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

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

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

Sample Response

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

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

```
{
   "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. 142).

The request accepts the following data in JSON format.

layerDigests (p. 16)

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. 16)

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. 16)

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

Type: String

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

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

Required: Yes

uploadId (p. 16)

The upload ID from a previous InitiateLayerUpload (p. 69) operation to associate with the image layer.

Type: String

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

Required: Yes

Response Syntax

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

Response Elements

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

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

layerDigest (p. 17)

The `sha256` digest of the image layer.

Type: String

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

registryId (p. 17)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 17)

The repository name associated with the request.

Type: String

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

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

uploadId (p. 17)

The upload ID associated with the layer.

Type: String

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

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
- 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

```
{
   "imageScanningConfiguration": {
      "scanOnPush": boolean
   },
   "imageTagMutability": "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. 142).

The request accepts the following data in JSON format.

imageScanningConfiguration (p. 20)

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. 129) object

Required: No

imageTagMutability (p. 20)

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

repositoryName (p. 20)

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

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

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

Required: Yes

tags (p. 20)

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. 141) objects

Required: No

Response Syntax

```
{
   "repository": {
      "createdAt": number,
      "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. 21)

The repository that was created.

Type: Repository (p. 139) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

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

Example

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: 283
Connection: keep-alive

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

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
- 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

registryId (p. 24)

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. 24)

The name of the repository.

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

lastEvaluatedAt (p. 24)

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

Type: Timestamp

lifecyclePolicyText (p. 24)

The JSON lifecycle policy text.

Type: String

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

registryId (p. 24)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 24)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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\":[{\"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
- 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. 142).

The request accepts the following data in JSON format.

force (p. 28)

If a repository contains images, forces the deletion.

Type: Boolean

Required: No

registryId (p. 28)

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. 28)

The name of the repository to delete.

Type: String

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

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

Required: Yes

Response Syntax

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

Type: Repository (p. 139) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

InvalidParameterException

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

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

Example

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
- AWS SDK for Java
- 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

registryId (p. 31)

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. 31)

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

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

policyText (p. 31)

The JSON repository policy that was deleted from the repository.

Type: String

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

registryId (p. 31)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 31)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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

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

See Also

For more information about using this API in one of the language-specific 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
- 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. 35).

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. 142).

The request accepts the following data in JSON format.

filter (p. 35)

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

Type: DescribeImagesFilter (p. 119) object

Required: No

imageIds (p. 35)

The list of image IDs for the requested repository.

Type: Array of ImageIdentifier (p. 125) objects

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

Required: No

maxResults (p. 35)

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. 35)

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. 35)

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. 35)

The repository that contains the images to describe.

Type: String

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

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

Required: Yes

Response Syntax

```
{
   "imageDetails": [
      {
         "imageDigest": "string",
         "imagePushedAt": number,
         "imageScanFindingsSummary": {
            "findingSeverityCounts": {
               "string" : number
            },
            "imageScanCompletedAt": number,
            "vulnerabilitySourceUpdatedAt": number
         },
         "imageScanStatus": {
            "description": "string",
            "status": "string"
         },
         "imageSizeInBytes": number,
         "imageTags": [ "string" ],
```

```
            "registryId": "string",
            "repositoryName": "string"
        }
    ],
    "nextToken": "string"
}
```

Response Elements

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

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

imageDetails (p. 36)

A list of ImageDetail (p. 122) objects that contain data about the image.

Type: Array of ImageDetail (p. 122) objects

nextToken (p. 36)

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. 144).

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

Example

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 `ubuntu` in the default account. Note that the image with the digest `sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ec8250899c397a` is tagged as `latest` and `xenial`.

Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 28
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.DescribeImages
X-Amz-Date: 20161216T193133Z
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"
}
```

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": [
    {
      "imageDigest":
 "sha256:7c70a5ebcc7fcaa22974a71175ba674efce3951fbec624943c891e9d256927c1",
      "imagePushedAt": 1452721263,
      "imageSizeInBytes": 44194573,
      "registryId": "012345678910",
      "repositoryName": "ubuntu"
    },
    {
      "imageDigest":
 "sha256:abdc090336ba4503bd72d0961a4f3d45134900d9a793d3f0c06a64d2555fbab7",
      "imagePushedAt": 1481916613,
      "imageSizeInBytes": 39142127,
      "imageTags": [
        "precise"
      ],
      "registryId": "012345678910",
```

```
      "repositoryName": "ubuntu"
    },
    {
      "imageDigest":
 "sha256:78dea3347768ba553ee8971cfb2dfa4c3efcd96db84ebfcdda59ba8d6057941b",
      "imagePushedAt": 1452721207,
      "imageSizeInBytes": 65747044,
      "registryId": "012345678910",
      "repositoryName": "ubuntu"
    },
    {
      "imageDigest":
 "sha256:7a64bc9c8843b0a8c8b8a7e4715b7615e4e1b0d8ca3c7e7a76ec8250899c397a",
      "imagePushedAt": 1481916523,
      "imageSizeInBytes": 50223482,
      "imageTags": [
        "xenial",
        "latest"
      ],
      "registryId": "012345678910",
      "repositoryName": "ubuntu"
    },
    {
      "imageDigest":
 "sha256:881befbe6f54c1e85029fe3a11554342bf765a0849600ecb8fa2f922798b4925",
      "imagePushedAt": 1481916647,
      "imageSizeInBytes": 65770577,
      "imageTags": [
        "trusty"
      ],
      "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
- 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. 142).

The request accepts the following data in JSON format.

imageId (p. 40)

An object with identifying information for an Amazon ECR image.

Type: ImageIdentifier (p. 125) object

Required: Yes

maxResults (p. 40)

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. 40)

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. 40)

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. 40)

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

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

imageId (p. 41)

An object with identifying information for an Amazon ECR image.

Type: ImageIdentifier (p. 125) object

imageScanFindings (p. 41)

The information contained in the image scan findings.

Type: ImageScanFindings (p. 127) object

imageScanStatus (p. 41)

The current state of the scan.

Type: ImageScanStatus (p. 130) object

nextToken (p. 41)

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. 41)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 41)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

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

Sample Response

```
HTTP/1.1 200 OK
```

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

{
    "imageScanFindings": {
        "findings": [
            {
                "name": "CVE-2019-5188",
                "description": "A code execution vulnerability exists in the directory
 rehashing functionality of E2fsprogs e2fsck 1.45.4. A specially crafted ext4 directory can
 cause an out-of-bounds write on the stack, resulting in code execution. An attacker can
 corrupt a partition to trigger this vulnerability.",
                "uri": "http://people.ubuntu.com/~ubuntu-security/cve/CVE-2019-5188",
                "severity": "MEDIUM",
                "attributes": [
                    {
                        "key": "package_version",
                        "value": "1.44.1-1ubuntu1.1"
                    },
                    {
                        "key": "package_name",
                        "value": "e2fsprogs"
                    },
                    {
                        "key": "CVSS2_VECTOR",
                        "value": "AV:L/AC:L/Au:N/C:P/I:P/A:P"
                    },
                    {
                        "key": "CVSS2_SCORE",
                        "value": "4.6"
                    }
                ]
            }
        ],
        "imageScanCompletedAt": 1579839105.0,
        "vulnerabilitySourceUpdatedAt": 1579811117.0,
        "findingSeverityCounts": {
            "MEDIUM": 1
        }
    },
    "registryId": "012345678910",
    "repositoryName": "sample-repo",
    "imageId": {
        "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
    },
    "imageScanStatus": {
        "status": "COMPLETE",
        "description": "The scan was completed successfully."
    }
}
```

See Also

For more information about using this API in one of the language-specific 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
- 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

```
{
   "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. 142).

The request accepts the following data in JSON format.

maxResults (p. 46)

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. 46)

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. 46)

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

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

Required: No

repositoryNames (p. 46)

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

Type: Array of strings

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

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

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

Required: No

Response Syntax

```
{
   "nextToken": "string",
   "repositories": [
      {
         "createdAt": number,
         "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. 47)

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. 47)

A list of repository objects corresponding to valid repositories.

Type: Array of Repository (p. 139) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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
Connection: keep-alive

{
   "repositories":[
      {
```

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


See Also

For more information about using this API in one of the language-specific 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
- 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

registryIds (p. 50)

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

```
{
   "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. 50)

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

Type: Array of AuthorizationData (p. 118) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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: 20151129T221940Z
User-Agent: aws-cli/1.9.9 Python/2.7.10 Darwin/14.5.0 botocore/1.3.9
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Sun, 29 Nov 2015 22:19:39 GMT
```

```
Content-Type: application/x-amz-json-1.1
Content-Length: 1590
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
  "authorizationData": [
    {
      "authorizationToken": "QVdTOkNpQzErSHF1ZXZPcUR...",
      "expiresAt": 1448878779.809,
      "proxyEndpoint": "https://012345678910.dkr.ecr.us-east-1.amazonaws.com"
    }
  ]
}
```

See Also

For more information about using this API in one of the language-specific 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
- 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

layerDigest (p. 53)

The digest of the image layer to download.

Type: String

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

Required: Yes

registryId (p. 53)

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. 53)

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

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

downloadUrl (p. 54)

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

Type: String

layerDigest (p. 54)

The digest of the image layer to download.

Type: String

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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
- 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. 142).

The request accepts the following data in JSON format.

registryId (p. 56)

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. 56)

The name of the repository.

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

lastEvaluatedAt (p. 56)

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

Type: Timestamp

lifecyclePolicyText (p. 56)

The JSON lifecycle policy text.

Type: String

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

registryId (p. 56)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 56)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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

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

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

Sample Response

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

{
   "lastEvaluatedAt":1.504295007E9,
   "lifecyclePolicyText":"{\"rules\":[{\"rulePriority\":1,\"description\":\"Expire
 images older than 14 days\",\"selection\":{\"tagStatus\":\"untagged\",\"countType\":
\"sinceImagePushed\",\"countUnit\":\"days\",\"countNumber\":14},\"action\":{\"type\":
\"expire\"}}]}",
   "registryId":"012345678910",
   "repositoryName":"project-a/amazon-ecs-sample"
}
```

See Also

For more information about using this API in one of the language-specific 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
- 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

```
{
   "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. 142).

The request accepts the following data in JSON format.

filter (p. 60)

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

Type: LifecyclePolicyPreviewFilter (p. 133) object

Required: No

imageIds (p. 60)

The list of imageIDs to be included.

Type: Array of ImageIdentifier (p. 125) objects

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

Required: No

maxResults (p. 60)

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. 60)

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. 60)

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. 60)

The name of the repository.

Type: String

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

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

Required: Yes

Response Syntax

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

```
   "status": "string",
   "summary": {
      "expiringImageTotalCount": number
   }
}
```

Response Elements

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

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

lifecyclePolicyText (p. 61)

The JSON lifecycle policy text.

Type: String

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

nextToken (p. 61)

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. 61)

The results of the lifecycle policy preview request.

Type: Array of LifecyclePolicyPreviewResult (p. 134) objects

registryId (p. 61)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 61)

The repository name associated with the request.

Type: String

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

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

status (p. 61)

The status of the lifecycle policy preview request.

Type: String

Valid Values: `IN_PROGRESS | COMPLETE | EXPIRED | FAILED`

summary (p. 61)

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

Type: LifecyclePolicyPreviewSummary (p. 136) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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: 20170901T222304Z
Authorization: AUTHPARAMS
Content-Length: 48

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 01 Sep 2017 22:23:06 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 640
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

registryId (p. 65)

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. 65)

The name of the repository with the policy to retrieve.

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

policyText (p. 65)

The JSON repository policy text associated with the repository.

Type: String

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

registryId (p. 65)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 65)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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

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

See Also

For more information about using this API in one of the language-specific 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
- 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. 142).

The request accepts the following data in JSON format.

registryId (p. 69)

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. 69)

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

Type: String

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

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

Required: Yes

Response Syntax

```
{
   "partSize": number,
   "uploadId": "string"
}
```

Response Elements

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

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

partSize (p. 69)

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

Type: Long

Valid Range: Minimum value of 0.

uploadId (p. 69)

The upload ID for the layer upload. This parameter is passed to further UploadLayerPart (p. 112) and CompleteLayerUpload (p. 16) 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. 144).

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

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
- 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. 7) 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. 142).

The request accepts the following data in JSON format.

filter (p. 72)

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

Type: ListImagesFilter (p. 138) object

Required: No

maxResults (p. 72)

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. 72)

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. 72)

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. 72)

The repository with image IDs to be listed.

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

imageIds (p. 73)

The list of image IDs for the requested repository.

Type: Array of ImageIdentifier (p. 125) objects

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

nextToken (p. 73)

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

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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
```

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

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

See Also

For more information about using this API in one of the language-specific 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
- 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

resourceArn (p. 76)

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

Type: String

Required: Yes

Response Syntax

```
{
   "tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Response Elements

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

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

tags (p. 76)

The tags for the resource.

Type: Array of Tag (p. 141) objects

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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
```

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

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
- 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

```
{
   "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. 142).

The request accepts the following data in JSON format.

imageManifest (p. 79)

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. 79)

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. 79)

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

Type: String

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

Required: No

registryId (p. 79)

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. 79)

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

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

image (p. 80)

Details of the image uploaded.

Type: Image (p. 120) object

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

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

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

Example

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\": \"sha256:5b52b314511a611975c2c65e695d920acdf8ae8848fe0ef00b7d018d1f118b64\"\n
   },\n   \"layers\": [\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 91768077,\n         \"digest
\": \"sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8\"\n      }\n
  ]\n}\n",
  "repositoryName": "amazonlinux",
  "imageTag": "2016.09"
}
```

Sample Response

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

{
  "image": {
    "imageId": {
      "imageDigest":
 "sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807",
      "imageTag": "2016.09"
    },
    "imageManifest": "{\n   \"schemaVersion\": 2,\n   \"mediaType\": \"application/
vnd.docker.distribution.manifest.v2+json\",\n   \"config\": {\n      \"mediaType\":
 \"application/vnd.docker.container.image.v1+json\",\n      \"size\": 1486,\n
 \"digest\": \"sha256:5b52b314511a611975c2c65e695d920acdf8ae8848fe0ef00b7d018d1f118b64\"\n
   },\n   \"layers\": [\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 91768077,\n         \"digest
\": \"sha256:8e3fa21c4cc40232e835a6761332d225c7af3235c5755f44ada2ed9d0e4ab7e8\"\n      }\n
  ]\n}\n",
    "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
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutImageScanningConfiguration

Updates the image scanning configuration for the specified repository.

Request Syntax

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

imageScanningConfiguration (p. 84)

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. 129) object

Required: Yes

registryId (p. 84)

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. 84)

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

Type: String

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

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

Required: Yes

Response Syntax

```
{
```

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

Response Elements

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

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

imageScanningConfiguration (p. 84)

The image scanning configuration setting for the repository.

Type: ImageScanningConfiguration (p. 129) object

registryId (p. 84)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 84)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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 scanning configuration for the `sample-repo` repository.

Sample Request

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
Content-Length: 90
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.PutImageScanningConfiguration
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",
    "imageScanningConfiguration": {
        "scanOnPush": true
    }
}
```

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: 114
Connection: keep-alive
x-amzn-RequestId: 3081a92b-2066-41f8-8a47-0580288ada9e

{
    "registryId": "012345678910",
    "repositoryName": "sample-repo",
    "imageScanningConfiguration": {
        "scanOnPush": true
    }
}
```

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
- 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. 142).

The request accepts the following data in JSON format.

imageTagMutability (p. 88)

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. 88)

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. 88)

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

Type: String

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

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

Required: Yes

Response Syntax

```
{
```

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

Response Elements

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

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

imageTagMutability (p. 88)

The image tag mutability setting for the repository.

Type: String

Valid Values: `MUTABLE | IMMUTABLE`

registryId (p. 88)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 88)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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
- 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. 142).

The request accepts the following data in JSON format.

lifecyclePolicyText (p. 92)

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. 92)

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. 92)

The name of the repository to receive the policy.

Type: String

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

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

Required: Yes

Response Syntax

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

```
}
```

Response Elements

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

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

lifecyclePolicyText (p. 92)

The JSON repository policy text.

Type: String

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

registryId (p. 92)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 92)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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

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\"}}]}",
   "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
- 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

```
{
   "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. 142).

The request accepts the following data in JSON format.

force (p. 96)

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. 96) operation. This is intended to prevent accidental repository lock outs.

Type: Boolean

Required: No

policyText (p. 96)

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. 96)

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. 96)

The name of the repository to receive the policy.

Type: String

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

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

Required: Yes

Response Syntax

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

Response Elements

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

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

policyText (p. 97)

The JSON repository policy text applied to the repository.

Type: String

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

registryId (p. 97)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 97)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

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

Example

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

```
POST / HTTP/1.1
Host: ecr.us-east-1.amazonaws.com
Accept-Encoding: identity
Content-Length: 223
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.SetRepositoryPolicy
X-Amz-Date: 20151214T235302Z
User-Agent: aws-cli/1.9.10 Python/2.7.10 Darwin/14.5.0 botocore/1.3.10
Content-Type: application/x-amz-json-1.1
Authorization: AWUTHPARAMS

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

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Mon, 14 Dec 2015 23:53:02 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 301
Connection: keep-alive
x-amzn-RequestId: cfc3ead9-a2bd-11e5-91c7-7126cb670c2b

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

```
}
```

See Also

For more information about using this API in one of the language-specific 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
- 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 day 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. 142).

The request accepts the following data in JSON format.

imageId (p. 100)

An object with identifying information for an Amazon ECR image.

Type: ImageIdentifier (p. 125) object

Required: Yes

registryId (p. 100)

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. 100)

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

Type: String

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

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

Required: Yes

Response Syntax

```
{
```

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

Response Elements

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

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

imageId (p. 100)

An object with identifying information for an Amazon ECR image.

Type: ImageIdentifier (p. 125) object

imageScanStatus (p. 100)

The current state of the scan.

Type: ImageScanStatus (p. 130) object

registryId (p. 100)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 100)

The repository name associated with the request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see Common Errors (p. 144).

ImageNotFoundException

The image requested does not exist in the specified repository.

HTTP Status Code: 400

InvalidParameterException

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

HTTP Status Code: 400

LimitExceededException

The operation did not succeed because it would have exceeded a service limit for your account. For more information, see Amazon ECR Service Quotas in the Amazon Elastic Container Registry User Guide.

HTTP Status Code: 400

RepositoryNotFoundException

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

HTTP Status Code: 400

ServerException

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

HTTP Status Code: 500

UnsupportedImageTypeException

The image is of a type that cannot be scanned.

HTTP Status Code: 400

Example

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

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

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

```
}
```

Sample Response

```
HTTP/1.1 200 OK
Server: Server
Date: Fri, 24 Jan 2020 03:48:07 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 208
Connection: keep-alive
x-amzn-RequestId: 3081a92b-2066-41f8-8a47-0580288ada9e

{
    "registryId": "012345678910",
    "repositoryName": "sample-repo",
    "imageId": {
        "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
    },
    "imageScanStatus": {
        "status": "IN_PROGRESS"
    }
}
```

See Also

For more information about using this API in one of the language-specific 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
- 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

lifecyclePolicyText (p. 104)

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. 104)

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. 104)

The name of the repository to be evaluated.

Type: String

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

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

Required: Yes

Response Syntax

```
{
```

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

Response Elements

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

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

lifecyclePolicyText (p. 104)

The JSON repository policy text.

Type: String

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

registryId (p. 104)

The registry ID associated with the request.

Type: String

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

repositoryName (p. 104)

The repository name associated with the request.

Type: String

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

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

status (p. 104)

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. 144).

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. Please try again later.

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

Example

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

```
POST / HTTP/1.1
Host: ecr.us-west-2.amazonaws.com
Accept-Encoding: identity
X-Amz-Target: AmazonEC2ContainerRegistry_V20150921.StartLifecyclePolicyPreview
Content-Type: application/x-amz-json-1.1
User-Agent: aws-cli/1.11.144 Python/3.6.1 Darwin/16.6.0 botocore/1.7.2
X-Amz-Date: 20170901T221604Z
Authorization: AUTHPARAMS
Content-Length: 535

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

Sample Response

```
HTTP/1.1 200 OK
```

```
Server: Server
Date: Fri, 01 Sep 2017 22:16:05 GMT
Content-Type: application/x-amz-json-1.1
Content-Length: 583
Connection: keep-alive
x-amzn-RequestId: 123a4b56-7c89-01d2-3ef4-example5678f

{
      "lifecyclePolicyText":"{\n    \"rules\": [\n        {\n            \"rulePriority
\": 1,\n            \"description\": \"Expire images older than 14 days\",\n
     \"selection\": {\n                \"tagStatus\": \"untagged\",\n
 \"countType\": \"sinceImagePushed\",\n                \"countUnit\": \"days\",\n
         \"countNumber\": 14\n            },\n            \"action\": {\n
 \"type\": \"expire\"\n            }\n        }\n    ]\n}\n",
      "registryId":"012345678910",
      "repositoryName":"project-a/amazon-ecs-sample",
      "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
- 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. 142).

The request accepts the following data in JSON format.

resourceArn (p. 108)

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. 108)

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. 141) 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. 144).

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
- 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

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

Request Parameters

For information about the parameters that are common to all actions, see Common Parameters (p. 142).

The request accepts the following data in JSON format.

resourceArn (p. 110)

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. 110)

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. 144).

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
- 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

```
{
   "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 (p. 142).

The request accepts the following data in JSON format.

layerPartBlob (p. 112)

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. 112)

The position of the first byte of the layer part witin the overall image layer.

Type: Long

Valid Range: Minimum value of 0.

Required: Yes

partLastByte (p. 112)

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. 112)

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. 112)

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

Type: String

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

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

Required: Yes

uploadId (p. 112)

The upload ID from a previous InitiateLayerUpload (p. 69) operation to associate with the layer part upload.

Type: String

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

Required: Yes

Response Syntax

```
{
   "lastByteReceived": number,
   "registryId": "string",
   "repositoryName": "string",
   "uploadId": "string"
}
```

Response Elements

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

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

lastByteReceived (p. 113)

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

Type: Long

Valid Range: Minimum value of 0.

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

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

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

uploadId (p. 113)

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. 144).

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

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
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The Amazon EC2 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

This data type is used in the ImageScanFinding (p. 126) data type.

Contents

key

The attribute key.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

value

The value assigned to the attribute key.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Required: No

See Also

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

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

The registry URL to use for this authorization token in a `docker login` command. The Amazon ECR registry URL format is `https://aws_account_id.dkr.ecr.region.amazonaws.com`. For example, `https://012345678910.dkr.ecr.us-east-1.amazonaws.com`..

Type: String

Required: No

See Also

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

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

DescribeImagesFilter

An object representing a filter on a DescribeImages (p. 35) operation.

Contents

tagStatus

The tag status with which to filter your DescribeImages (p. 35) 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
- AWS SDK for Ruby V3

Image

An object representing an Amazon ECR image.

Contents

imageId

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

Type: ImageIdentifier (p. 125) 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 media type associated with the image manifest.

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

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
- AWS SDK for Ruby V3

ImageDetail

An object that describes an image returned by a DescribeImages (p. 35) operation.

Contents

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

imageScanFindingsSummary

A summary of the last completed image scan.

Type: ImageScanFindingsSummary (p. 128) object

Required: No

imageScanStatus

The current state of the scan.

Type: ImageScanStatus (p. 130) 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. 35).

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

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
- 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`

Required: No

failureReason

The reason for the failure.

Type: String

Required: No

imageId

The image ID associated with the failure.

Type: ImageIdentifier (p. 125) 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
- AWS SDK for Ruby V3

ImageIdentifier

An object with identifying information for an Amazon ECR image.

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
- 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. 117) 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
- AWS SDK for Ruby V3

ImageScanFindings

The details of an image scan.

Contents

findings

The findings from the image scan.

Type: Array of ImageScanFinding (p. 126) 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
- 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
- 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 StartImageScan (p. 100) 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
- 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`

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
- 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.gzip` or `application/vnd.oci.image.layer.v1.tar+gzip`.

Type: String

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- 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
- 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
- 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. 137) 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
- 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
- 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
- AWS SDK for Ruby V3

ListImagesFilter

An object representing a filter on a ListImages (p. 72) operation.

Contents

tagStatus

The tag status with which to filter your ListImages (p. 72) 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
- 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

imageScanningConfiguration

The image scanning configuration for a repository.

Type: ImageScanningConfiguration (p. 129) 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

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

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

Required: No

repositoryUri

The URI for the repository. You can use this URI for Docker `push` or `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
- AWS SDK for Ruby V3

Tag

The metadata that you apply to a resource 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.

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: No

Value

The optional part of a key-value pair that make up a tag. A `value` acts as a descriptor within a tag category (key).

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
- 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 YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

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

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

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

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

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

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

Type: string

Required: Conditional

X-Amz-Date

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

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is

not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see Handling Dates in Signature Version 4 in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

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

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

Type: string

Required: Conditional

X-Amz-Signature

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

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

Type: string

Required: Conditional

X-Amz-SignedHeaders

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

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

Type: string

Required: Conditional

Common Errors

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

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

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

HTTP Status Code: 500

InvalidAction

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

HTTP Status Code: 400

InvalidClientTokenId

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

HTTP Status Code: 403

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

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

HTTP Status Code: 400

InvalidQueryParameter

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

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

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

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

OptInRequired

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

HTTP Status Code: 403

RequestExpired

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

HTTP Status Code: 400

ServiceUnavailable

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

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

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

HTTP Status Code: 400