Amazon Web Services

General Reference

Version 1.0

Amazon Web Services: General Reference

Table of Contents

AWS General Reference

This is the AWS Documentation General Reference. It covers the following topics:

AWS Regions and Endpoints

To reduce data latency in your applications, most Amazon Web Services offer a regional endpoint to make your requests. An endpoint is a URL that is the entry point for a web service. For example, `https://dynamodb.us-west-2.amazonaws.com` is an entry point for the Amazon DynamoDB service.

Some services, such as IAM, do not support regions; therefore, their endpoints do not include a region. Some services, such as Amazon EC2, let you specify an endpoint that does not include a specific region, for example, `https://ec2.amazonaws.com`. In that case, AWS routes the endpoint to us-east-1.

If a service supports regions, the resources in each region are independent. For example, if you create an Amazon EC2 instance or an Amazon SQS queue in one region, the instance or queue is independent from instances or queues in another region.

To see the supported services per region in a tabbed format, see the Region Table. This page does not include endpoint information.

For information about which regions and endpoints are supported for each service, see the following tables.

Topics

Amazon API Gateway

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	apigateway.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	apigateway.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	apigateway.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	apigateway.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	apigateway.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	apigateway.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	apigateway.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	apigateway.ap-northeast-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
EU (Frankfurt)	eu-central-1	apigateway.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	apigateway.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	apigateway.eu-west-2.amazonaws.com	HTTPS

Application Auto Scaling

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	autoscaling.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	autoscaling.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	autoscaling.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	autoscaling.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	autoscaling.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	autoscaling.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	autoscaling.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	autoscaling.ap-southeast-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	autoscaling.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	autoscaling.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	autoscaling.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	autoscaling.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	autoscaling.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	autoscaling.sa-east-1.amazonaws.com	HTTP and HTTPS

Amazon AppStream

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	appstream.us-east-1.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	appstream.ap-northeast-1.amazonaws.com	HTTPS

Amazon AppStream 2.0

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	appstream2.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	appstream2.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	appstream2.eu-west-1.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	appstream2.ap-northeast-1.amazonaws.com	HTTPS

Auto Scaling

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	autoscaling.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	autoscaling.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	autoscaling.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	autoscaling.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	autoscaling.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	autoscaling.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	autoscaling.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	autoscaling.ap-southeast-1.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Sydney)	ap-southeast-2	autoscaling.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	autoscaling.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	autoscaling.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	autoscaling.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	autoscaling.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	autoscaling.sa-east-1.amazonaws.com	HTTP and HTTPS

If you just specify the general endpoint (autoscaling.amazonaws.com), Auto Scaling directs your request to the us-east-1 endpoint.

For information about using Auto Scaling in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS Batch

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	batch.us-east-1.amazonaws.com	HTTPS

Athena

Region Name	Region	Endpoint	Connection String	Protocol
US East (N. Virginia)	us-east-1	athena.us-east-1.amazonaws.com	jdbc:awsathena://athena.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	athena.us-west-2.amazonaws.com	jdbc:awsathena://athena.us-west-2.amazonaws.com	HTTPS

Note
Use the Connection String to connect to the service via JDBC (Java Database Connectivity) driver.

AWS Certificate Manager

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	acm.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	acm.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	acm.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	acm.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	acm.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	acm.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	acm.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	acm.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	acm.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	acm.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	acm.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	acm.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	acm.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	acm.sa-east-1.amazonaws.com	HTTPS

AWS CloudFormation

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cloudformation.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	cloudformation.us-east-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US West (N. California)	us-west-1	cloudformation.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cloudformation.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	cloudformation.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	cloudformation.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	cloudformation.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	cloudformation.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cloudformation.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	cloudformation.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	cloudformation.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	cloudformation.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	cloudformation.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	cloudformation.sa-east-1.amazonaws.com	HTTPS

For information about using AWS CloudFormation in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon CloudFront

Amazon CloudFront distributions have a single endpoint: cloudfront.amazonaws.com and only supports HTTPS requests. When you submit requests to CloudFront programmatically, specify us-east-1 for the US East (N. Virginia) Region.

The CloudFront hosted zone ID value is `Z2FDTNDATAQYW2`.

AWS CloudHSM

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cloudhsm.us-east-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US East (Ohio)	us-east-2	cloudhsm.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	cloudhsm.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cloudhsm.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	cloudhsm.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	cloudhsm.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cloudhsm.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	cloudhsm.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	cloudhsm.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	cloudhsm.eu-west-1.amazonaws.com	HTTPS

For information about using AWS CloudHSM in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon CloudSearch

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cloudsearch.us-east-1.amazonaws.com	HTTPS
US West (N. California)	us-west-1	cloudsearch.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cloudsearch.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	cloudsearch.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	cloudsearch.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cloudsearch.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	cloudsearch.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	cloudsearch.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	cloudsearch.eu-west-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
South America (São Paulo)	sa-east-1	cloudsearch.sa-east-1.amazonaws.com	HTTPS

AWS CloudTrail

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cloudtrail.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	cloudtrail.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	cloudtrail.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cloudtrail.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	cloudtrail.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	cloudtrail.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	cloudtrail.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	cloudtrail.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cloudtrail.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	cloudtrail.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	cloudtrail.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	cloudtrail.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	cloudtrail.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	cloudtrail.sa-east-1.amazonaws.com	HTTPS

For information about using AWS CloudTrail in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon CloudWatch

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	monitoring.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	monitoring.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	monitoring.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	monitoring.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	monitoring.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	monitoring.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	monitoring.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	monitoring.ap-southeast-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	monitoring.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	monitoring.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	monitoring.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	monitoring.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	monitoring.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	monitoring.sa-east-1.amazonaws.com	HTTP and HTTPS

For information about using Amazon CloudWatch in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon CloudWatch Events

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	events.us-east-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US East (Ohio)	us-east-2	events.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	events.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	events.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	events.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	events.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	events.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	events.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	events.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	events.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	events.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	events.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	events.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	events.sa-east-1.amazonaws.com	HTTPS

Amazon CloudWatch Logs

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	logs.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	logs.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	logs.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	logs.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	logs.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	logs.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	logs.ap-northeast-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Singapore)	ap-southeast-1	logs.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	logs.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	logs.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	logs.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	logs.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	logs.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	logs.sa-east-1.amazonaws.com	HTTPS

For information about using Amazon CloudWatch Logs in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS CodeBuild

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	codebuild.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	codebuild.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	codebuild.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	codebuild.eu-west-1.amazonaws.com	HTTPS

AWS CodeCommit

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	codecommit.us-east-1.amazonaws.com	HTTPS and SSH
US East (Ohio)	us-east-2	codecommit.us-east-2.amazonaws.com	HTTPS and SSH
US West (Oregon)	us-west-2	codecommit.us-west-2.amazonaws.com	HTTPS and SSH
EU (Ireland)	eu-west-1	codecommit.eu-west-1.amazonaws.com	HTTPS and SSH

For information about Git connection endpoints, see Regions and Git Connection Endpoints for AWS CodeCommit.

AWS CodeDeploy

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	codedeploy.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	codedeploy.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	codedeploy.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	codedeploy.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	codedeploy.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	codedeploy.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	codedeploy.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	codedeploy.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	codedeploy.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	codedeploy.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	codedeploy.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	codedeploy.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	codedeploy.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	codedeploy.sa-east-1.amazonaws.com	HTTPS

AWS CodePipeline

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	codepipeline.us-east-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US East (Ohio)	us-east-2	codepipeline.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	codepipeline.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	codepipeline.eu-west-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	codepipeline.eu-central-1.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	codepipeline.ap-northeast-1.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	codepipeline.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	codepipeline.ap-southeast-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	codepipeline.sa-east-1.amazonaws.com	HTTPS

Amazon Cognito Identity

Amazon Cognito Identity includes Amazon Cognito Your User Pools and Amazon Cognito Federated Identities.

Amazon Cognito Your User Pools

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cognito-idp.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	cognito-idp.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cognito-idp.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	cognito-idp.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cognito-idp.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	cognito-idp.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	cognito-idp.eu-central-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
EU (Ireland)	eu-west-1	cognito-idp.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	cognito-idp.eu-west-2.amazonaws.com	HTTPS

Amazon Cognito Federated Identities

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cognito-identity.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	cognito-identity.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cognito-identity.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	cognito-identity.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cognito-identity.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	cognito-identity.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	cognito-identity.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	cognito-identity.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	cognito-identity.eu-west-2.amazonaws.com	HTTPS

Amazon Cognito Sync

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	cognito-sync.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	cognito-sync.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	cognito-sync.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	cognito-sync.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	cognito-sync.ap-southeast-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Tokyo)	ap-northeast-1	cognito-sync.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	cognito-sync.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	cognito-sync.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	cognito-sync.eu-west-2.amazonaws.com	HTTPS

AWS Config

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	config.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	config.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	config.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	config.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	config.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	config.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	config.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	config.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	config.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	config.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	config.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	config.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	config.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	config.sa-east-1.amazonaws.com	HTTPS

For information about using AWS Config in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS Config Rules

You can use AWS Config rules to evaluate your AWS resource configurations in the following regions.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	config.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	config.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	config.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	config.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	config.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	config.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	config.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	config.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	config.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	config.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	config.eu-west-2.amazonaws.com	HTTPS

AWS Data Pipeline

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	datapipeline.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	datapipeline.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	datapipeline.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	datapipeline.ap-northeast-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	datapipeline.eu-west-1.amazonaws.com	HTTPS

AWS Database Migration Service

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	dms.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	dms.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	dms.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	dms.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	dms.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	dms.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	dms.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	dms.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	dms.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	dms.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	dms.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	dms.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	dms.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	dms.sa-east-1.amazonaws.com	HTTPS

AWS Device Farm

Region Name	Region	Endpoint	Protocol
US West (Oregon)	us-west-2	devicefarm.us-west-2.amazonaws.com	HTTPS

Amazon DevPay

Region Name	Region	Endpoint	Protocol
n/a	n/a	ls.amazonaws.com	HTTPS

AWS Direct Connect

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	directconnect.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	directconnect.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	directconnect.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	directconnect.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	directconnect.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	directconnect.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	directconnect.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	directconnect.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	directconnect.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	directconnect.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	directconnect.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	directconnect.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	directconnect.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	directconnect.sa-east-1.amazonaws.com	HTTPS

AWS Directory Service

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	ds.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	ds.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	ds.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	ds.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	ds.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ds.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ds.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ds.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	ds.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	ds.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	ds.eu-west-2.amazonaws.com	HTTPS

Only AWS Directory Service for Microsoft Active Directory (Enterprise Edition) is available in EU (Frankfurt).

Amazon DynamoDB

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	dynamodb.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	dynamodb.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	dynamodb.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	dynamodb.us-west-2.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
Canada (Central)	ca-central-1	dynamodb.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	dynamodb.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	dynamodb.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	dynamodb.ap-southeast-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	dynamodb.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	dynamodb.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	dynamodb.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	dynamodb.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	dynamodb.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	dynamodb.sa-east-1.amazonaws.com	HTTP and HTTPS

For information about using Amazon DynamoDB in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon DynamoDB Streams

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	streams.dynamodb.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	streams.dynamodb.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	streams.dynamodb.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	streams.dynamodb.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	streams.dynamodb.ca-central-1.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Mumbai)	ap-south-1	streams.dynamodb.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	streams.dynamodb.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	streams.dynamodb.ap-southeast-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	streams.dynamodb.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	streams.dynamodb.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	streams.dynamodb.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	streams.dynamodb.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	streams.dynamodb.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	streams.dynamodb.sa-east-1.amazonaws.com	HTTP and HTTPS

For information about using Amazon DynamoDB Streams in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon EC2 Container Registry

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	ecr.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	ecr.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	ecr.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	ecr.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	ecr.ca-central-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	ecr.eu-central-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
EU (Ireland)	eu-west-1	ecr.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	ecr.eu-west-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ecr.ap-northeast-1.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ecr.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ecr.ap-southeast-2.amazonaws.com	HTTPS

Amazon EC2 Container Service

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	ecs.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	ecs.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	ecs.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	ecs.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	ecs.ca-central-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	ecs.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	ecs.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	ecs.eu-west-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ecs.ap-northeast-1.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ecs.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ecs.ap-southeast-2.amazonaws.com	HTTPS

Amazon EC2 Systems Manager

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	ssm.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	ssm.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	ssm.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	ssm.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ssm.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ssm.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ssm.ap-northeast-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	ssm.ap-northeast-2.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	ssm.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	ssm.eu-west-1.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	ssm.sa-east-1.amazonaws.com	HTTPS

AWS Elastic Beanstalk

Region Name	Region	Endpoint	Protocol	Amazon Route 53 Hosted Zone ID
US East (N. Virginia)	us-east-1	elasticbeanstalk.us-east-1.amazonaws.com	HTTPS	Z117KPS5GTRQ2G
US East (Ohio)	us-east-2	elasticbeanstalk.us-east-2.amazonaws.com	HTTPS	Z14LCN19Q5QHIC
US West (N. California)	us-west-1	elasticbeanstalk.us-west-1.amazonaws.com	HTTPS	Z1LQECGX5PH1X
US West (Oregon)	us-west-2	elasticbeanstalk.us-west-2.amazonaws.com	HTTPS	Z38NKT9BP95V3O
Canada (Central)	ca-central-1	elasticbeanstalk.ca-central-1.amazonaws.com	HTTPS	ZJFCZL7SSZB5I

Region Name	Region	Endpoint	Protocol	Amazon Route 53 Hosted Zone ID
Asia Pacific (Mumbai)	ap-south-1	elasticbeanstalk.ap-south-1.amazonaws.com	HTTPS	Z18NTBI3Y7N9TZ
Asia Pacific (Seoul)	ap-northeast-2	elasticbeanstalk.ap-northeast-2.amazonaws.com	HTTPS	Z3JE5OI70TWKCP
Asia Pacific (Singapore)	ap-southeast-1	elasticbeanstalk.ap-southeast-1.amazonaws.com	HTTPS	Z16FZ9L249IFLT
Asia Pacific (Sydney)	ap-southeast-2	elasticbeanstalk.ap-southeast-2.amazonaws.com	HTTPS	Z2PCDNR3VC2G1N
Asia Pacific (Tokyo)	ap-northeast-1	elasticbeanstalk.ap-northeast-1.amazonaws.com	HTTPS	Z1R25G3KIG2GBW
EU (Frankfurt)	eu-central-1	elasticbeanstalk.eu-central-1.amazonaws.com	HTTPS	Z1FRNW7UH4DEZJ
EU (Ireland)	eu-west-1	elasticbeanstalk.eu-west-1.amazonaws.com	HTTPS	Z2NYPWQ7DFZAZH
EU (London)	eu-west-2	elasticbeanstalk.eu-west-2.amazonaws.com	HTTPS	Z1GKAAAUGATPF1
South America (São Paulo)	sa-east-1	elasticbeanstalk.sa-east-1.amazonaws.com	HTTPS	Z10X7K2B4QSOFV

AWS Elastic Beanstalk Health Service

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	elasticbeanstalk-health.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	elasticbeanstalk-health.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	elasticbeanstalk-health.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	elasticbeanstalk-health.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	elasticbeanstalk-health.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	elasticbeanstalk-health.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	elasticbeanstalk-health.ap-northeast-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Singapore)	ap-southeast-1	elasticbeanstalk-health.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	elasticbeanstalk-health.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	elasticbeanstalk-health.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	elasticbeanstalk-health.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	elasticbeanstalk-health.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	elasticbeanstalk-health.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	elasticbeanstalk-health.sa-east-1.amazonaws.com	HTTPS

Amazon Elastic Compute Cloud (Amazon EC2)

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	ec2.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	ec2.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	ec2.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	ec2.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	ec2.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	ec2.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	ec2.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ec2.ap-southeast-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ec2.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ec2.ap-northeast-1.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
EU (Frankfurt)	eu-central-1	ec2.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	ec2.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	ec2.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	ec2.sa-east-1.amazonaws.com	HTTP and HTTPS

For information about using Amazon EC2 in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Elastic File System

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	elasticfilesystem.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	elasticfilesystem.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	elasticfilesystem.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	elasticfilesystem.eu-west-1.amazonaws.com	HTTPS

Elastic Load Balancing

Region Name	Region	Endpoint	Protocol	Amazon Route 53 Hosted Zone ID
US East (N. Virginia)	us-east-1	elasticloadbalancing.us-east-1.amazonaws.com	HTTPS	Z35SXDOTRQ7X7K
US East (Ohio)	us-east-2	elasticloadbalancing.us-east-2.amazonaws.com	HTTPS	Z3AADJGX6KTTL2
US West (N. California)	us-west-1	elasticloadbalancing.us-west-1.amazonaws.com	HTTPS	Z368ELLRRE2KJ0
US West (Oregon)	us-west-2	elasticloadbalancing.us-west-2.amazonaws.com	HTTPS	Z1H1FL5HABSF5
Canada (Central)	ca-central-1	elasticloadbalancing.ca-central-1.amazonaws.com	HTTPS	ZQSVJUPU6J1EY

Region Name	Region	Endpoint	Protocol	Amazon Route 53 Hosted Zone ID
Asia Pacific (Mumbai)	ap-south-1	elasticloadbalancing.ap-south-1.amazonaws.com	HTTPS	ZP97RAFLXTNZK
Asia Pacific (Seoul)	ap-northeast-2	elasticloadbalancing.ap-northeast-2.amazonaws.com	HTTPS	ZWKZPGTI48KDX
Asia Pacific (Singapore)	ap-southeast-1	elasticloadbalancing.ap-southeast-1.amazonaws.com	HTTPS	Z1LMS91P8CMLE5
Asia Pacific (Sydney)	ap-southeast-2	elasticloadbalancing.ap-southeast-2.amazonaws.com	HTTPS	Z1GM3OXH4ZPM65
Asia Pacific (Tokyo)	ap-northeast-1	elasticloadbalancing.ap-northeast-1.amazonaws.com	HTTPS	Z14GRHDCWA56QT
EU (Frankfurt)	eu-central-1	elasticloadbalancing.eu-central-1.amazonaws.com	HTTPS	Z215JYRZR1TBD5
EU (Ireland)	eu-west-1	elasticloadbalancing.eu-west-1.amazonaws.com	HTTPS	Z32O12XQLNTSW2
EU (London)	eu-west-2	elasticloadbalancing.eu-west-2.amazonaws.com	HTTPS	ZHURV8PSTC4K8
South America (São Paulo)	sa-east-1	elasticloadbalancing.sa-east-1.amazonaws.com	HTTPS	Z2P70J7HTTTPLU

If you just specify the general endpoint (elasticloadbalancing.amazonaws.com), Elastic Load Balancing directs your request to the us-east-1 endpoint.

For information about using Elastic Load Balancing in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Elastic Transcoder

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	elastictranscoder.us-east-1.amazonaws.com	HTTPS
US West (N. California)	us-west-1	elastictranscoder.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	elastictranscoder.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	elastictranscoder.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	elastictranscoder.ap-southeast-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Sydney)	ap-southeast-2	elastictranscoder.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	elastictranscoder.ap-northeast-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	elastictranscoder.eu-west-1.amazonaws.com	HTTPS

Amazon ElastiCache

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	elasticache.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	elasticache.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	elasticache.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	elasticache.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	elasticache.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	elasticache.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	elasticache.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	elasticache.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	elasticache.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	elasticache.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	elasticache.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	elasticache.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	elasticache.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	elasticache.sa-east-1.amazonaws.com	HTTPS

For information about using Amazon ElastiCache in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Elasticsearch Service

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	us-east-1.es.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	us-east-2.es.amazonaws.com	HTTPS
US West (N. California)	us-west-1	us-west-1.es.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	us-west-2.es.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	ap-south-1.es.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	ap-northeast-2.es.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ap-southeast-1.es.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ap-southeast-2.es.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ap-northeast-1.es.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	eu-central-1.es.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	eu-west-1.es.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	sa-east-1.es.amazonaws.com	HTTPS

Amazon EMR

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	elasticmapreduce.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	elasticmapreduce.us-east-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US West (N. California)	us-west-1	elasticmapreduce.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	elasticmapreduce.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	elasticmapreduce.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	elasticmapreduce.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	elasticmapreduce.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	elasticmapreduce.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	elasticmapreduce.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	elasticmapreduce.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	elasticmapreduce.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	elasticmapreduce.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	elasticmapreduce.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	elasticmapreduce.sa-east-1.amazonaws.com	HTTPS

If you specify the general endpoint (elasticmapreduce.amazonaws.com), Amazon EMR directs your request to an endpoint in the default region. For accounts created on or after March 8, 2013, the default region is us-west-2; for older accounts, the default region is us-east-1.

For information about using Amazon EMR in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon GameLift

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	gamelift.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	gamelift.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	gamelift.ap-south-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Seoul)	ap-northeast-2	gamelift.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	gamelift.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	gamelift.ap-northeast-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	gamelift.eu-west-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	gamelift.eu-central-1.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	gamelift.sa-east-1.amazonaws.com	HTTPS

Amazon Glacier

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	glacier.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	glacier.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	glacier.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	glacier.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	glacier.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	glacier.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	glacier.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	glacier.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	glacier.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	glacier.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	glacier.eu-west-1.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
EU (London)	eu-west-2	glacier.eu-west-2.amazonaws.com	HTTP and HTTPS

For information about using Amazon Glacier in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS Health

AWS Health has a single endpoint: health.us-east-1.amazonaws.com (HTTPS).

AWS Identity and Access Management (IAM)

IAM has a single endpoint: https://iam.amazonaws.com.

For information about using AWS Identity and Access Management in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS Import/Export

AWS Snowball is a standalone service now. For region information on that service, see AWS Snowball (p. 56).

AWS Import/Export Disk

AWS Import/Export Disk has a single endpoint for all regions.

Endpoint	Protocol
importexport.amazonaws.com	HTTPS

Amazon Inspector

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	inspector.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	inspector.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	inspector.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	inspector.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	inspector.ap-southeast-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Tokyo)	ap-northeast-1	inspector.ap-northeast-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	inspector.eu-west-1.amazonaws.com	HTTPS

AWS IoT

The following table provides a list of region-specific endpoints that AWS IoT supports for working with rules, certificates, and policies.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	iot.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	iot.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	iot.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	iot.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	iot.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	iot.ap-northeast-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	iot.ap-northeast-2.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	iot.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	iot.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	iot.eu-west-2.amazonaws.com	HTTPS

The following table provides a list of region-specific endpoints that AWS IoT supports for working with Thing Shadows. To look up your account-specific prefix, use the describe-endpoint command.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	*prefix*.iot.us-east-1.amazonaws.com	HTTPS, MQTT
US East (Ohio)	us-east-2	*prefix*.iot.us-east-2.amazonaws.com	HTTPS, MQTT
US West (Oregon)	us-west-2	*prefix*.iot.us-west-2.amazonaws.com	HTTPS, MQTT
Asia Pacific (Singapore)	ap-southeast-1	*prefix*.iot.ap-southeast-1.amazonaws.com	HTTPS, MQTT

Region Name	Region	Endpoint	Protocol
Asia Pacific (Sydney)	ap-southeast-2	*prefix*.iot.ap-southeast-2.amazonaws.com	HTTPS, MQTT
Asia Pacific (Tokyo)	ap-northeast-1	*prefix*.iot.ap-northeast-1.amazonaws.com	HTTPS, MQTT
Asia Pacific (Seoul)	ap-northeast-2	*prefix*.iot.ap-northeast-2.amazonaws.com	HTTPS, MQTT
EU (Frankfurt)	eu-central-1	*prefix*.iot.eu-central-1.amazonaws.com	HTTPS, MQTT
EU (Ireland)	eu-west-1	*prefix*.iot.eu-west-1.amazonaws.com	HTTPS, MQTT
EU (London)	eu-west-2	*prefix*.iot.eu-west-2.amazonaws.com	HTTPS, MQTT

AWS IoT supports multiple protocols for accessing the message broker and the Thing Shadows component. The following table lists the ports to use for each protocol.

Port	Protocol	Authentication Mechanism
443	HTTPS	Signature Version 4
443	MQTT over WebSocket	Signature Version 4
8443	HTTPS	TLS client authentication, with certificates
8883	MQTT	TLS client authentication, with certificates

AWS Key Management Service

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	kms.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	kms.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	kms.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	kms.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	kms.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	kms.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	kms.ap-northeast-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Singapore)	ap-southeast-1	kms.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	kms.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	kms.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	kms.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	kms.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	kms.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	kms.sa-east-1.amazonaws.com	HTTPS

For information about using AWS Key Management Service in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Kinesis Analytics

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	kinesisanalytics.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	kinesisanalytics.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	kinesisanalytics.eu-west-1.amazonaws.com	HTTPS

Amazon Kinesis Firehose

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	firehose.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	firehose.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	firehose.eu-west-1.amazonaws.com	HTTPS

Amazon Kinesis Streams

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	kinesis.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	kinesis.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	kinesis.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	kinesis.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	kinesis.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	kinesis.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	kinesis.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	kinesis.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	kinesis.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	kinesis.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	kinesis.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	kinesis.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	kinesis.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	kinesis.sa-east-1.amazonaws.com	HTTPS

AWS Lambda

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	lambda.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	lambda.us-east-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US West (N. California)	us-west-1	lambda.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	lambda.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	lambda.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	lambda.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	lambda.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	lambda.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	lambda.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	lambda.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	lambda.eu-west-2.amazonaws.com	HTTPS

Amazon Lex

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	runtime.lex.us-east-1.amazonaws.com	HTTPS

Amazon Lightsail

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	lightsail.us-east-1.amazonaws.com	HTTPS

Amazon Machine Learning

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	machinelearning.us-east-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
EU (Ireland)	eu-west-1	machinelearning.eu-west-1.amazonaws.com	HTTPS

Amazon Mechanical Turk

Region	Endpoint	Protocol
Sandbox endpoint for Amazon Mechanical Turk actions.	mechanicalturk.sandbox.amazonaws.com	HTTPS
Production endpoint for Amazon Mechanical Turk actions.	mechanicalturk.amazonaws.com	HTTPS

Amazon Mobile Analytics

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	mobileanalytics.us-east-1.amazonaws.com	HTTPS

AWS OpsWorks

AWS OpsWorks uses the following regional endpoints.

AWS OpsWorks for Chef Automate

You can create and manage AWS OpsWorks for Chef Automate servers in the following regions. Resources can be managed only in the region in which they are created. Resources that are created in one regional endpoint are not available, nor can they be cloned to, another regional endpoint.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia) Region	us-east-1	opsworks-cm.us-east-1.amazonaws.com	HTTPS
US West (Oregon) Region	us-west-2	opsworks-cm.us-west-2.amazonaws.com	HTTPS
EU (Ireland) Region	eu-west-1	opsworks-cm.eu-west-1.amazonaws.com	HTTPS

AWS OpsWorks Stacks

You can create and manage AWS OpsWorks resources in all regions except AWS GovCloud (US) and the China (Beijing) Region. Resources can be managed only in the region in which they are created. Resources that are created in one regional endpoint are not available, nor can they be cloned to, another regional endpoint.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	opsworks.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	opsworks.us-east-2.amazonaws.com	HTTPS
US West (N. California) Region	us-west-1	opsworks.us-west-1.amazonaws.com	HTTPS
US West (Oregon) Region	us-west-2	opsworks.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo) Region	ap-northeast-1	opsworks.ap-northeast-1.amazonaws.com	HTTPS
Asia Pacific (Seoul) Region	ap-northeast-2	opsworks.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	opsworks.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Singapore) Region	ap-southeast-1	opsworks.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney) Region	ap-southeast-2	opsworks.ap-southeast-2.amazonaws.com	HTTPS
EU (Frankfurt) Region	eu-central-1	opsworks.eu-central-1.amazonaws.com	HTTPS
EU (Ireland) Region	eu-west-1	opsworks.eu-west-1.amazonaws.com	HTTPS
EU (London) Region	eu-west-2	opsworks.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo) Region	sa-east-1	opsworks.sa-east-1.amazonaws.com	HTTPS

AWS Organizations

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	organizations.us-east-1.amazonaws.com	HTTPS

Amazon Pinpoint

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	pinpoint.us-east-1.amazonaws.com	HTTPS

Amazon Polly

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	polly.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	polly.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	polly.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	polly.eu-west-1.amazonaws.com	HTTPS

Amazon Redshift

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	redshift.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	redshift.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	redshift.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	redshift.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	redshift.ca-central-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Mumbai)	ap-south-1	redshift.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	redshift.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	redshift.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	redshift.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	redshift.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	redshift.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	redshift.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	redshift.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	redshift.sa-east-1.amazonaws.com	HTTPS

For information about using Amazon Redshift in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Rekognition

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	rekognition.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	rekognition.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	rekognition.eu-west-1.amazonaws.com	HTTPS

Amazon Relational Database Service (Amazon RDS)

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US East (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS

For information about using Amazon Relational Database Service in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Route 53

Amazon Route 53 uses two endpoints. The endpoint that you use depends on the operation that you want to perform.

Requests for hosted zones, resource record sets, health checks, and cost allocation tags use the following endpoint.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	route53.amazonaws.com	HTTPS

Requests for domain registration use the following endpoint.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	route53domains.us-east-1.amazonaws.com	HTTPS

AWS Security Token Service (AWS STS)

The default endpoint for AWS Security Token Service is https://sts.amazonaws.com, which serves all global requests. You can also make calls to other regional endpoints that are activated for your AWS account. All regions are activated by default, but you can deactivate regions that you do not intend to use. If you deactivate a region, you must reactivate it for your account in the AWS Management Console before you can use that region's endpoint.

For more information, see Activating and Deactivating AWS STS in an AWS Region in the *IAM User Guide*.

Region Name	Region	Endpoint	Protocol
--Global--	--Global--	sts.amazonaws.com	HTTPS
US East (N. Virginia)	us-east-1	sts.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	sts.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	sts.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	sts.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	sts.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	sts.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	sts.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	sts.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	sts.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	sts.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	sts.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	sts.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	sts.eu-west-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
South America (São Paulo)	sa-east-1	sts.sa-east-1.amazonaws.com	HTTPS

For information about using AWS Security Token Service in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS Service Catalog

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	servicecatalog.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	servicecatalog.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	servicecatalog.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	servicecatalog.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	servicecatalog.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	servicecatalog.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	servicecatalog.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	servicecatalog.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	servicecatalog.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	servicecatalog.eu-west-2.amazonaws.com	HTTPS

AWS Shield Advanced

AWS Shield Advanced has a single endpoint: shield.us-east-1.amazonaws.com. It supports HTTPS requests only.

Amazon Simple Email Service (Amazon SES)

Region Name	Region	API (HTTPS) Endpoint	SMTP Endpoint	Email Sending or Receiving
US East (N. Virginia)	us-east-1	email.us-east-1.amazonaws.com	email-smtp.us-east-1.amazonaws.com	Email sending
US West (Oregon)	us-west-2	email.us-west-2.amazonaws.com	email-smtp.us-west-2.amazonaws.com	Email sending
EU (Ireland)	eu-west-1	email.eu-west-1.amazonaws.com	email-smtp.eu-west-1.amazonaws.com	Email sending
US East (N. Virginia)	us-east-1	N/A	inbound-smtp.us-east-1.amazonaws.com	Email receiving
US West (Oregon)	us-west-2	N/A	inbound-smtp.us-west-2.amazonaws.com	Email receiving
EU (Ireland)	eu-west-1	N/A	inbound-smtp.eu-west-1.amazonaws.com	Email receiving

Amazon Simple Notification Service (Amazon SNS)

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	sns.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	sns.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	sns.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	sns.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	sns.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	sns.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	sns.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	sns.ap-southeast-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	sns.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	sns.ap-northeast-1.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
EU (Frankfurt)	eu-central-1	sns.eu-central-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	sns.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	sns.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	sns.sa-east-1.amazonaws.com	HTTP and HTTPS

For information about using Amazon Simple Notification Service in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Simple Queue Service (Amazon SQS)

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	sqs.us-east-1.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	sqs.us-east-2.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	sqs.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	sqs.us-west-2.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	sqs.ca-central-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	sqs.ap-south-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	sqs.ap-northeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	sqs.ap-southeast-1.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	sqs.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	sqs.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	sqs.eu-central-1.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
EU (Ireland)	eu-west-1	sqs.eu-west-1.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	sqs.eu-west-2.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	sqs.sa-east-1.amazonaws.com	HTTP and HTTPS

For information about using Amazon Simple Queue Service in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon SQS Legacy Endpoints

If you use the AWS CLI or SDK for Python, you can use the following legacy endpoints.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	queue.amazonaws.com	HTTP and HTTPS
US East (Ohio)	us-east-2	us-east-2.queue.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	us-west-1.queue.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	us-west-2.queue.amazonaws.com	HTTP and HTTPS
Canada (Central)	ca-central-1	ca-central-1.queue.amazonaws.com	HTTP and HTTPS
Asia Pacific (Mumbai)	ap-south-1	ap-south-1.queue.amazonaws.com	HTTP and HTTPS
Asia Pacific (Seoul)	ap-northeast-2	ap-northeast-2.queue.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ap-southeast-1.queue.amazonaws.com	HTTP and HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ap-southeast-2.queue.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ap-northeast-1.queue.amazonaws.com	HTTP and HTTPS
EU (Frankfurt)	eu-central-1	eu-central-1.queue.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	eu-west-1.queue.amazonaws.com	HTTP and HTTPS
EU (London)	eu-west-2	eu-west-2.queue.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
South America (São Paulo)	sa-east-1	sa-east-1.queue.amazonaws.com	HTTP and HTTPS

Amazon Simple Storage Service (Amazon S3)

When sending requests to these endpoints using the REST API, you can use the virtual-hosted style and path-style methods. For more information, see Virtual Hosting of Buckets.

Note

Amazon S3 renamed the US Standard Region to the US East (N. Virginia) Region to be consistent with AWS regional naming conventions. There is no change to the endpoint and you do not need to make any changes to your application.

Region Name	Region	Endpoint	Location Constraint	Protocol	Signature Version(s) Support
US East (N. Virginia)	us-east-1	Valid endpoint names for this region: • s3.amazonaws.com • s3-external-1.amazonaws.com • s3.dualstack.us-east-1.amazonaws.com**	(none required)	HTTP and HTTPS	Versions 2 and 4
US East (Ohio)	us-east-2	Valid endpoint names for this region: • s3.us-east-2.amazonaws.com • s3-us-east-2.amazonaws.com • s3.dualstack.us-east-2.amazonaws.com	us-east-2	HTTP and HTTPS	Versions 4 only
US West (N. California)	us-west-1	Valid endpoint names for this region: • s3-us-west-1.amazonaws.com • s3.dualstack.us-west-1.amazonaws.com**	us-west-1	HTTP and HTTPS	Versions 2 and 4
US West (Oregon)	us-west-2	Valid endpoint names for this region: • s3-us-west-2.amazonaws.com • s3.dualstack.us-west-2.amazonaws.com**	us-west-2	HTTP and HTTPS	Versions 2 and 4
Canada (Central)	ca-central-1	Valid endpoint names for this region: • s3.ca-central-1.amazonaws.com • s3-ca-central-1.amazonaws.com	ca-central-1	HTTP and HTTPS	Version 4 only

Region Name	Region	Endpoint	Location Constraint	Protocol	Signature Version(s) Support
		• s3.dualstack.ca-central-1.amazonaws.com**			
Asia Pacific (Mumbai)	ap-south-1	Valid endpoint names for this region: • s3.ap-south-1.amazonaws.com • s3-ap-south-1.amazonaws.com • s3.dualstack.ap-south-1.amazonaws.com**	ap-south-1	HTTP and HTTPS	Version 4 only
Asia Pacific (Seoul)	ap-northeast-2	Valid endpoint names for this region: • s3.ap-northeast-2.amazonaws.com • s3-ap-northeast-2.amazonaws.com • s3.dualstack.ap-northeast-2.amazonaws.com**	ap-northeast-2	HTTP and HTTPS	Version 4 only
Asia Pacific (Singapore)	ap-southeast-1	Valid endpoint names for this region • s3-ap-southeast-1.amazonaws.com • s3.dualstack.ap-southeast-1.amazonaws.com**	ap-southeast-1	HTTP and HTTPS	Versions 2 and 4
Asia Pacific (Sydney)	ap-southeast-2	Valid endpoint names for this region: • s3-ap-southeast-2.amazonaws.com • s3.dualstack.ap-southeast-2.amazonaws.com**	ap-southeast-2	HTTP and HTTPS	Versions 2 and 4
Asia Pacific (Tokyo)	ap-northeast-1	Valid endpoint names for this region: • s3-ap-northeast-1.amazonaws.com • s3.dualstack.ap-northeast-1.amazonaws.com**	ap-northeast-1	HTTP and HTTPS	Versions 2 and 4

Region Name	Region	Endpoint	Location Constraint	Protocol	Signature Version(s) Support
EU (Frankfurt)	eu-central-1	Valid endpoint names for this region: • s3.eu-central-1.amazonaws.com • s3-eu-central-1.amazonaws.com • s3.dualstack.eu-central-1.amazonaws.com**	eu-central-1	HTTP and HTTPS	Version 4 only
EU (Ireland)	eu-west-1	Valid endpoint names for this region: • s3-eu-west-1.amazonaws.com • s3.dualstack.eu-west-1.amazonaws.com**	EU or eu-west-1	HTTP and HTTPS	Versions 2 and 4
EU (London)	eu-west-2	Valid endpoint names for this region: • s3.eu-west-2.amazonaws.com • s3-eu-west-2.amazonaws.com • s3.dualstack.eu-west-2.amazonaws.com**	eu-west-2	HTTP and HTTPS	Version 4 only
South America (São Paulo)	sa-east-1	Valid endpoint names for this region: • s3-sa-east-1.amazonaws.com • s3.dualstack.sa-east-1.amazonaws.com**	sa-east-1	HTTP and HTTPS	Versions 2 and 4

Note

**Amazon S3 dual-stack endpoints support requests to S3 buckets over IPv6 and IPv4. For more information, see Using Dual-Stack Endpoints.

Important

If you use a region other than the US East (N. Virginia) endpoint to create a bucket, you must set the `LocationConstraint` bucket parameter to the same region. Both the AWS SDK for Java and AWS SDK for .NET use an enumeration for setting location constraints (`Region` for Java, `S3Region` for .NET). For more information, see PUT Bucket in the *Amazon Simple Storage Service API Reference*.

Amazon Simple Storage Service Website Endpoints

When you configure your bucket as a website, the website is available using the following region-specific website endpoints. Note that the website endpoints are different than the REST API endpoints listed in the preceding table. For more information about hosting websites on Amazon S3, see Hosting Websites on Amazon S3 in the *Amazon Simple Storage Service Developer Guide*. You need the hosted zone IDs when using the Amazon Route 53 API to add an alias record to your hosted zone.

Note
The website endpoints do not support https.

Region Name	Website Endpoint	Amazon Route 53 Hosted Zone ID
US East (N. Virginia)	s3-website-us-east-1.amazonaws.com	Z3AQBSTGFYJSTF
US East (Ohio)	s3-website.us-east-2.amazonaws.com	Z2O1EMRO9K5GLX
US West (N. California)	s3-website-us-west-1.amazonaws.com	Z2F56UZL2M1ACD
US West (Oregon)	s3-website-us-west-2.amazonaws.com	Z3BJ6K6RIION7M
Canada (Central)	s3-website.ca-central-1.amazonaws.com	Z1QDHH18159H29
Asia Pacific (Mumbai)	s3-website.ap-south-1.amazonaws.com	Z11RGJOFQNVJUP
Asia Pacific (Seoul)	s3-website.ap-northeast-2.amazonaws.com	Z3W03O7B5YMIYP
Asia Pacific (Singapore)	s3-website-ap-southeast-1.amazonaws.com	Z3O0J2DXBE1FTB
Asia Pacific (Sydney)	s3-website-ap-southeast-2.amazonaws.com	Z1WCIGYICN2BYD
Asia Pacific (Tokyo)	s3-website-ap-northeast-1.amazonaws.com	Z2M4EHUR26P7ZW
EU (Frankfurt)	s3-website.eu-central-1.amazonaws.com	Z21DNDUVLTQW6Q
EU (Ireland)	s3-website-eu-west-1.amazonaws.com	Z1BKCTXD74EZPE
EU (London)	s3-website.eu-west-2.amazonaws.com	Z3GKZC51ZF0DB4
South America (São Paulo)	s3-website-sa-east-1.amazonaws.com	Z7KQH4QJS55SO

For information about using Amazon Simple Storage Service in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon Simple Workflow Service (Amazon SWF)

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	swf.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	swf.us-east-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US West (N. California)	us-west-1	swf.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	swf.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	swf.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	swf.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	swf.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	swf.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	swf.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	swf.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	swf.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	swf.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	swf.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	swf.sa-east-1.amazonaws.com	HTTPS

For information about using Amazon Simple Workflow Service in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

Amazon SimpleDB

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	sdb.amazonaws.com	HTTP and HTTPS
US West (N. California)	us-west-1	sdb.us-west-1.amazonaws.com	HTTP and HTTPS
US West (Oregon)	us-west-2	sdb.us-west-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Singapore)	ap-southeast-1	sdb.ap-southeast-1.amazonaws.com	HTTP and HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Sydney)	ap-southeast-2	sdb.ap-southeast-2.amazonaws.com	HTTP and HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	sdb.ap-northeast-1.amazonaws.com	HTTP and HTTPS
EU (Ireland)	eu-west-1	sdb.eu-west-1.amazonaws.com	HTTP and HTTPS
South America (São Paulo)	sa-east-1	sdb.sa-east-1.amazonaws.com	HTTP and HTTPS

AWS Snowball

AWS Snowball, used with a standard Snowball appliance, is available in the following regions and includes these endpoints.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	snowball.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	snowball.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	snowball.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	snowball.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	snowball.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	snowball.ap-southeast-2.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	snowball.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	snowball.eu-west-1.amazonaws.com	HTTPS

AWS Snowball, used with an AWS Snowball Edge appliance, is available in the following regions and includes these endpoints.

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	snowball.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	snowball.us-east-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US West (N. California)	us-west-1	snowball.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	snowball.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	snowball.ap-southeast-2.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	snowball.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	snowball.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	snowball.eu-west-2.amazonaws.com	HTTPS

For information about using AWS Snowball in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS Step Functions

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	states.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	states.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	states.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	states.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	states.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	states.eu-west-1.amazonaws.com	HTTPS

AWS Storage Gateway

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	storagegateway.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	storagegateway.us-east-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
US West (N. California)	us-west-1	storagegateway.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	storagegateway.us-west-2.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	storagegateway.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	storagegateway.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	storagegateway.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	storagegateway.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	storagegateway.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	storagegateway.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	storagegateway.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	storagegateway.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	storagegateway.sa-east-1.amazonaws.com	HTTPS

AWS Support

AWS Support has a single endpoint: support.us-east-1.amazonaws.com (HTTPS).

Amazon VPC

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	ec2.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	ec2.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	ec2.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	ec2.us-west-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Canada (Central)	ca-central-1	ec2.ca-central-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	ec2.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	ec2.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	ec2.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	ec2.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	ec2.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	ec2.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	ec2.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	ec2.eu-west-2.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	ec2.sa-east-1.amazonaws.com	HTTPS

If you specify the general endpoint (ec2.amazonaws.com), Amazon VPC directs your request to the us-east-1 endpoint.

For information about using Amazon VPC in AWS GovCloud (US), see AWS GovCloud (US) Endpoints.

AWS WAF

AWS WAF for CloudFront distributions has a single endpoint: waf.amazonaws.com. It supports HTTPS requests only.

AWS WAF for Application Load Balancers has the following endpoints:

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	waf-regional.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	waf-regional.us-west-2.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	waf-regional.eu-west-1.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	waf-regional.ap-northeast-1.amazonaws.com	HTTPS

Amazon WorkMail

Region Name	Region	Service	Endpoint
US East (N. Virginia)	us-east-1	Autodiscover	autodiscover-service.mail.us-east-1.awsapps.com
US East (N. Virginia)	us-east-1	Exchange Web Service	ews.mail.us-east-1.awsapps.com
US East (N. Virginia)	us-east-1	Exchange Active Sync	mobile.mail.us-east-1.awsapps.com
US East (N. Virginia)	us-east-1	MAPI Proxy	outlook.mail.us-east-1.awsapps.com
US East (N. Virginia)	us-east-1	IMAPS	imap.mail.us-east-1.awsapps.com
US West (Oregon)	us-west-2	Autodiscover	autodiscover-service.mail.us-west-2.awsapps.com
US West (Oregon)	us-west-2	Exchange Web Service	ews.mail.us-west-2.awsapps.com
US West (Oregon)	us-west-2	Exchange Active Sync	mobile.mail.us-west-2.awsapps.com
US West (Oregon)	us-west-2	MAPI Proxy	outlook.mail.us-west-2.awsapps.com
US West (Oregon)	us-west-2	IMAPS	imap.mail.us-west-2.awsapps.com
EU (Ireland)	eu-west-1	Autodiscover	autodiscover-service.mail.eu-west-1.awsapps.com
EU (Ireland)	eu-west-1	Exchange Web Service	ews.mail.eu-west-1.awsapps.com
EU (Ireland)	eu-west-1	Exchange Active Sync	mobile.mail.eu-west-1.awsapps.com
EU (Ireland)	eu-west-1	MAPI Proxy	outlook.mail.eu-west-1.awsapps.com
EU (Ireland)	eu-west-1	IMAPS	imap.mail.eu-west-1.awsapps.com

Amazon WorkSpaces

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	workspaces.us-east-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	workspaces.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	workspaces.ap-southeast-1.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
Asia Pacific (Sydney)	ap-southeast-2	workspaces.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	workspaces.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	workspaces.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	workspaces.eu-west-1.amazonaws.com	HTTPS

AWS X-Ray

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	xray.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	xray.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	xray.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	xray.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	xray.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	xray.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	xray.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	xray.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	xray.ap-northeast-1.amazonaws.com	HTTPS
EU (Frankfurt)	eu-central-1	xray.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	xray.eu-west-1.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	xray.sa-east-1.amazonaws.com	HTTPS

AWS Security Credentials

When you interact with AWS, you specify your AWS *security credentials* to verify who you are and whether you have permission to access the resources that you are requesting. AWS uses the security credentials to authenticate and authorize your requests.

For example, if you want to download a specific file from an Amazon Simple Storage Service (Amazon S3) bucket, your credentials must allow that access. If your credentials aren't authorized to download the file, AWS denies your request.

> **Note**
> In some cases, you can make calls to AWS without security credentials, such as downloading a file that is publicly shared in an Amazon S3 bucket.

Topics

Root Account Credentials vs. IAM User Credentials

All AWS accounts have root account credentials (that is, the credentials of the account owner). These credentials allow full access to all resources in the account. Because you can't restrict permissions for root account credentials, we recommend that you delete your root access keys and then create AWS Identity and Access Management (IAM) user credentials for everyday interaction with AWS. For more information, see Lock away your AWS account (root) access keys in the *IAM User Guide*.

> **Note**
> You may need root account access for specific tasks, such as changing a AWS support plan or closing your account. In these cases, sign in to the AWS Management Console with your email and password. See Email and password (account root user) (p. 64).

For a list of tasks that require account root user access, see AWS Tasks that Require Account Root User (p. 63).

With IAM, you can securely control access to AWS services and resources for users in your AWS account. For example, if you require administrator-level permissions, you can create an IAM user, grant that user full access, and then use those credentials to interact with AWS. If you need to modify or revoke your permissions, you can delete or modify the policies that are associated with that IAM user.

If you have multiple users that require access to your AWS account, you can create unique credentials for each user and define who has access to which resources. You don't need to share credentials. For example, you can create IAM users with read-only access to resources in your AWS account and distribute those credentials to your users.

> **Note**
> Any activity or costs that are associated with the IAM user are billed to the AWS account owner.

AWS Tasks that Require Account Root User

The tasks listed below require you to sign in as the account root user. We normally recommend that you use a standard IAM user with appropriate permissions to perform all normal user or administrative tasks. However, you can perform the tasks listed below only when you sign in as the root user of an account.

- Modify root user details (p. 64). This include changing the root user's password.
- Change your AWS support plan.
- Change or delete your payment options.
- View your account's billing information. For information about how to enable billing access for IAM users see Activating Access to the Billing and Cost Management Console
- Close an AWS account.
- Sign up for GovCloud.
- Submit a Reverse DNS for Amazon EC2 request. The "this form" link on that page to submit a request works only if you sign in with root creds.
- Create a CloudFront key pair.
- Create an AWS-created X.509 signing certificate. (You can still make self-created certificates for IAM users.)
- Transfer an Amazon Route 53 domain to another AWS account.
- Change the Amazon EC2 setting for longer resource IDs. Changing the setting as root affects all users and roles in the account. Changing as an IAM user or IAM role affects only that user or role.
- Submit a request to perform penetration testing on your AWS infrastructure.

Understanding and Getting Your Security Credentials

You use different types of security credentials depending on how you interact with AWS. For example, you use a user name and password to sign in to the AWS Management Console. You use access keys to make programmatic calls to AWS API actions.

If you forget or lose your credentials, you can't recover them. For security reasons, AWS doesn't allow you to retrieve your passwords or secret access keys and does not store the private keys that are part of a key pair. However, you can create new credentials and then disable or delete the old credentials.

Note

Security credentials are account specific. If you have access to multiple AWS accounts, use the credentials that are associated with the account that you want to access.

Getting AWS root account credentials is different than getting IAM user credentials. For AWS root account credentials, you get credentials, such as access keys or key pairs, from the Security Credentials page in the AWS Management Console. For IAM user credentials, you get credentials from the IAM console.

The following list describes the types of AWS security credentials, when you might use them, and how to get each type of credential for the AWS root account or for an IAM user.

Topics

Email and password (account root user)

When you sign up for AWS, you provide an email address and password that is associated with your AWS account. You use these credentials to sign in to AWS web pages such as the AWS Management Console, AWS discussion forums, or AWS support center. The account email address and password are root-level credentials, and anyone who uses these credentials has full access to all resources in the account. We recommend that you can use an IAM user name and password to sign in to AWS web pages. For more information, see Root Account Credentials vs. IAM User Credentials (p. 62).

The email address and password are specified when the AWS account was created. You can change the email address and password on the Security Credentials page. You can also choose **Forgot your password?** on the AWS sign in page to reset your password.

IAM user name and password

When multiple individuals or applications require access to your AWS account, AWS Identity and Access Management (IAM) lets you create unique IAM user identities. Users can use their own user names and passwords to sign in to the AWS Management Console, AWS discussion forums, or AWS support center. In some cases, an IAM user name and password are required to use a service, such as sending email with SMTP by using Amazon Simple Email Service (Amazon SES).

For more information about IAM users, see Identities (Users, Groups, and Roles) in the *IAM User Guide*.

You specify user names when you create them. After you create users, you can create passwords for each user. For more information, see Managing Passwords for IAM Users in the *IAM User Guide*.

Note

IAM users can manage their own password but only if they have been given permission. For more information, see Permitting IAM Users to Change Their Own Password in the *IAM User Guide*.

Multi-Factor Authentication (MFA)

AWS Multi-Factor Authentication (AWS MFA) provides an extra level of security that you can apply to your AWS account. With AWS MFA enabled, when you sign in to an AWS website, you are prompted for your user name and password, and an authentication code from an MFA device. Together, they provide increased security for your AWS account settings and resources.

By default, MFA (multi-factor authentication) is not enabled. You can enable and manage MFA devices for the AWS root account by going to the Security Credentials page or the IAM dashboard in the AWS Management Console. For more information about enabling MFA for IAM users, see Enabling MFA Devices in the *IAM User Guide*.

> **Note**
> For additional security, we recommend that you require MFA on the root account credentials and highly privileged IAM users. For more information, see Using Multi-Factor Authentication (MFA) in AWS in the *IAM User Guide*.

Access keys (access key ID and secret access key)

Access keys consist of an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). You use access keys to sign programmatic requests that you make to AWS if you use the AWS SDKs, REST, or Query APIs. The AWS SDKs use your access keys to sign requests for you, so that you don't have to handle the signing process. You can also sign requests manually. For more information, see Signing AWS API Requests (p. 98).

Access keys are also used with command line interfaces (CLIs). When you use a CLI, the commands that you issue are signed by your access keys, which you can either pass with the command or store as configuration settings on your computer.

You can also create and use temporary access keys, known as *temporary security credentials*. In addition to the access key ID and secret access key, temporary security credentials include a security token that you must send to AWS when you use temporary security credentials. The advantage of temporary security credentials is that they are short-term. After they expire, they're no longer valid. You can use temporary access keys in less secure environments or distribute them to grant users temporary access to resources in your AWS account. For example, you can grant entities from other AWS accounts access to resources in your AWS account (cross-account access) or grant users who don't have AWS security credentials access to resources in your AWS account (federation). For more information, see Temporary Security Credentials in the *IAM User Guide*.

You can have a maximum of two access keys (active or inactive) at a time. For your AWS (root) account, see Managing Access Keys for your AWS Account (p. 69). For IAM users, you can create IAM access keys with the IAM console. For more information, see Creating, Modifying, and Viewing Access Keys (AWS Management Console) in the *IAM User Guide*.

> **Important**
> If you or your IAM users forget or lose the secret access key, you can create a new access key.

Key pairs

Key pairs consist of a public key and a private key. You use the private key to create a digital signature, and then AWS uses the corresponding public key to validate the signature. Key pairs are used only for Amazon EC2 and Amazon CloudFront.

For Amazon EC2, you use key pairs to access Amazon EC2 instances, such as when you use SSH to log in to a Linux instance. For more information, see Connect to Your Linux Instances in the *Amazon EC2 User Guide for Linux Instances*.

For Amazon CloudFront, you use key pairs to create signed URLs for private content, such as when you want to distribute restricted content that someone paid for. For more information, see Serving Private Content through CloudFront in the *Amazon CloudFront Developer Guide*.

AWS does not provide key pairs for your account; you must create them. You can create Amazon EC2 key pairs from the Amazon EC2 console, CLI, or API. For more information, see Amazon EC2 Key Pairs in the *Amazon EC2 User Guide for Linux Instances*.

You create Amazon CloudFront key pairs from the Security Credentials page. Only the root account (not IAM users) can create CloudFront key pairs. For more information, see Serving Private Content through CloudFront in the *Amazon CloudFront Developer Guide*.

AWS Account Identifiers

AWS assigns two unique IDs to each AWS account:

- An AWS account ID
- A canonical user ID

The AWS account ID is a 12-digit number, such as 123456789012, that you use to construct Amazon Resource Names (ARNs). When you refer to resources, such as an IAM user or an Amazon Glacier vault, the account ID distinguishes your resources from resources in other AWS accounts.

The canonical user ID is a long string, such as `79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be`.

You can use canonical user IDs in an Amazon S3 bucket policy for cross-account access, which means an AWS account can access resources in another AWS account. For example, to grant another AWS account access to your bucket, you specify the account's canonical user ID in the bucket's policy. For more information, see Bucket Policy Examples in the *Amazon Simple Storage Service Developer Guide*.

Finding Your Account Identifiers

For AWS account users (root account users), you can get both IDs from the **Account Identifiers** section of the Security Credentials page. You can't change either ID.

For IAM or federated users, you can get your AWS account ID from the Support Center dashboard. You can also choose **Support** and then choose **Support Center**. The ID is displayed on the upper right. The account ID for an AWS account is the same for the root account and its IAM users. For more information, see Your AWS Account ID and Its Alias.

> **Note**
> You can also return the canonical user ID with the Amazon S3 `ListBuckets` API. For more information, see GET Service Response Elements in the *Amazon Simple Storage Service API Reference*.

Best Practices for Managing AWS Access Keys

When you access AWS programmatically, you use an access key to verify your identity and the identity of your applications. An access key consists of an access key ID (something like `AKIAIOSFODNN7EXAMPLE`) and a secret access key (something like `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`).

Anyone who has your access key has the same level of access to your AWS resources that you do. Consequently, AWS goes to significant lengths to protect your access keys, and, in keeping with our shared-responsibility model, you should as well.

The steps that follow can help you protect access keys. For general background, see AWS Security Credentials (p. 62).

> **Note**
> Your organization may have different security requirements and policies than those described in this topic. The suggestions provided here are intended to be general guidelines.

Topics

Remove (or Don't Generate) a Root Account Access Key

An access key is required in order to sign requests that you make using the AWS Command Line Tools, the AWS SDKs, or direct API calls. Anyone who has the access key for your root account has unrestricted access to all the resources in your account, including billing information. You cannot restrict the permissions for your root account.

One of the best ways to protect your account is to not have an access key for your root account. Unless you *must* have a root access key (which is very rare), it is best not to generate one. Instead, the recommended best practice is to create one or more AWS Identity and Access Management (IAM) users, give them the necessary permissions, and use IAM users for everyday interaction with AWS.

If you already have an access key for your account, we recommend that you find places in your applications where you are currently using that key (if any), replace the root access key with an IAM user access key, and then disable and remove the root access key. For details about how to substitute one access key for another, see the post How to Rotate Access Keys for IAM Users on the AWS Security Blog.

By default, AWS does not generate an access key for new accounts.

For information about how to create an IAM user with administrative permissions, see Creating Your First IAM Admin User and Group in the *IAM User Guide*.

Use Temporary Security Credentials (IAM Roles) Instead of Long-Term Access Keys

In many scenarios, you don't need a long-term access key that never expires (as you have with an IAM user). Instead, you can create IAM roles and generate temporary security credentials. Temporary security credentials consist of an access key ID and a secret access key, but they also include a security token that indicates when the credentials expire.

Long-term access keys, such as those associated with IAM users and AWS accounts (root), remain valid until you manually revoke them. However, temporary security credentials obtained through IAM roles and other features of the AWS Security Token Service expire after a short period of time. Use temporary security credentials to help reduce your risk in case credentials are accidentally exposed.

Use an IAM role and temporary security credentials in these scenarios:

- **You have an application or AWS CLI scripts running on an Amazon EC2 instance.** Do not pass an access key to the application, embed it in the application, or have the application read a key from a source such as an Amazon S3 bucket (even if the bucket is encrypted). Instead, define an IAM role that has appropriate permissions for your application and launch the Amazon EC2 instance with roles for EC2. This associates an IAM role with the Amazon EC2 instance and lets the application get temporary security credentials that it can in turn use to make AWS calls. The AWS SDKs and the AWS CLI can get temporary credentials from the role automatically.
- **You need to grant cross-account access.** Use an IAM role to establish trust between accounts, and then grant users in one account limited permissions to access the trusted account. For more information, see Tutorial: Delegate Access Across AWS Accounts Using IAM Roles in the *IAM User Guide*.

- **You have a mobile app.** Do not embed an access key with the app, even in encrypted storage. Instead, use Amazon Cognito to manage user identity in your app. This service lets you authenticate users using Login with Amazon, Facebook, Google, or any OpenID Connect (OIDC)–compatible identity provider. You can then use the Amazon Cognito credentials provider to manage credentials that your app uses to make requests to AWS. For more information, see Using the Amazon Cognito Credentials Provider on the AWS Mobile Blog.
- **You want to federate into AWS and your organization supports SAML 2.0.** If you work for an organization that has an identity provider that supports SAML 2.0, configure the provider to use SAML to exchange authentication information with AWS and get back a set of temporary security credentials. For more information, see About SAML 2.0-based Federation in the *IAM User Guide*.
- **You want to federate into AWS and your organization has an on-premises identity store.** If users can authenticate inside your organization, you can write an application that can issue them temporary security credentials for access to AWS resources. For more information, see Creating a URL that Enables Federated Users to Access the AWS Management Console (Custom Federation Broker) in the *IAM User Guide*.

Manage IAM User Access Keys Properly

If you do need to create access keys for programmatic access to AWS, create an IAM user and grant that user only the permissions he or she needs. Then generate an access key for that user. For details, see Managing Access Keys for IAM Users in the *IAM User Guide*.

> **Note**
> Remember that if you are running an application on an Amazon EC2 instance and the application needs access to AWS resources, you should use IAM roles for EC2, as described in the previous section.

Observe these precautions when using access keys:

- **Don't embed access keys directly into code.** The AWS SDKs and the AWS Command Line Tools allow you to put access keys in known locations so that you do not have to keep them in code.

  Put access keys in one of the following locations:

  - **The AWS credentials file.** The AWS SDKs and AWS CLI automatically use the credentials that you store in the AWS credentials file.

    For information about using the AWS credentials file, see the documentation for your SDK. Examples include Set up AWS Credentials and Region for Development in the *AWS SDK for Java Developer Guide* and Configuration and Credential Files in the *AWS Command Line Interface User Guide*.

    > **Note**
    > To store credentials for the AWS SDK for .NET and the AWS Tools for Windows PowerShell, we recommend you use the SDK Store. For more information, see Using the SDK Store in the *AWS SDK for .NET Developer Guide*.
  - **Environment variables.** On a multitenant system, choose user environment variables, not system environment variables.

    For more information about using environment variables to store credentials, see Environment Variables in the *AWS Command Line Interface User Guide*.
- **Use different access keys for different applications.** Do this so that you can isolate the permissions and revoke the access keys for individual applications if an access key is exposed. Having separate access keys for different applications also generates distinct entries in AWS CloudTrail log files, which makes it easier for you to determine which application performed specific actions.
- **Rotate access keys periodically.** Change access keys on a regular basis. For details, see Rotating Access Keys (AWS CLI, Tools for Windows PowerShell, and AWS API) in the *IAM User Guide* and How to Rotate Access Keys for IAM Users on the AWS Security Blog.

- **Remove unused access keys.** If a user leaves your organization, remove the corresponding IAM user so that the user's access to your resources is removed. To find out when an access key was last used, use the `GetAccessKeyLastUsed` API (AWS CLI command: `aws iam get-access-key-last-used`).
- **Configure multifactor authentication for your most sensitive operations.** For details, see Using Multi-Factor Authentication (MFA) in AWS in the *IAM User Guide*.

More Resources

For more information about best practices for keeping your AWS account secure, see the following resources:

- IAM Best Practices. This topic presents a list of suggestions for using the AWS Identity and Access Management (IAM) service to help secure your AWS resources.
- The following pages provide guidance for setting up the AWS SDKs and the AWS CLI to use access keys.
  - Set up AWS Credentials and Region for Development in the *AWS SDK for Java Developer Guide*.
  - Using the SDK Store in the *AWS SDK for .NET Developer Guide*.
  - Providing Credentials to the SDK in the *AWS SDK for PHP Developer Guide*.
  - Configuration in the Boto 3 (AWS SDK for Python) documentation.
  - Using AWS Credentials in the *AWS Tools for Windows PowerShell* guide.
  - Configuration and Credential Files in the *AWS Command Line Interface User Guide*.
- Granting Access Using an IAM Role. This walkthrough discusses how programs written using the .NET SDK can automatically get temporary security credentials when running on an Amazon EC2 instance. A similar topic is available for the AWS SDK for Java.

Managing Access Keys for your AWS Account

You can create, rotate, disable, or delete access keys (access key IDs and secret access keys) for your AWS account (root user). Anyone who has an access key for your AWS account has unrestricted access to all the resources in your account, including billing information.

We recommend that you don't create access keys for your AWS account and delete any that exist. Instead, create a user in AWS Identity and Access Management (IAM) and choose **Programmatic access** to create an access key for the user. For more information, see Lock away your AWS account (root) access keys in the *IAM User Guide*.

When you create an access key, AWS gives you an opportunity to view and download the secret access key only once. If you don't download it or if you lose it, you can delete the access key and then create a new one.

A newly created access key has the status of *active*, which means that you can use the access key for API calls. You can have up to two access keys for your AWS account, which is useful when you want to rotate the access keys (p. 68). When you disable an access key, you can't use it for API calls.

You can create or delete an access key any time. However, when you delete an access key, it's gone forever and can't be retrieved.

Creating, Disabling, and Deleting Access Keys for your AWS Account

Follow these steps to manage access keys for your AWS account. For information about managing access keys for IAM users, see Managing Access Keys for IAM Users in the *IAM User Guide*.

To create, disable, or delete an access key for your AWS (root) account

1. Use your AWS account email address and password to sign in to the AWS Management Console.

   If you previously signed in to the console with IAM user credentials, your browser might open your IAM user sign-in page. You can't use the IAM user sign-in page to sign in with your AWS account credentials. Instead, choose **Sign-in using root account credentials** to go to the AWS account sign-in page.
2. In the top right of the console, choose your account name or number. Then choose **My Security Credentials**.
3. Choose **Continue to Security Credentials**.
4. Expand the **Access Keys (Access Key ID and Secret Access Key)** section.
5. Choose your desired action.

   To create an access key

   Choose **Create New Access Key**. Then choose **Download Key File** to save the access key ID and secret access key to a file on your computer. After you close the dialog box, you can't retrieve this secret access key again.

   To disable an existing access key

   Choose **Make Inactive** next to the access key that you are disabling. To reenable an inactive access key, choose **Make Active**.

   To delete an existing access key

   Before you delete an access key, make sure it's no longer in use. For more information, see Finding unused access keys in the *IAM User Guide*. You can't recover an access key after deleting it. Then, choose **Delete** next to the access key that you are deleting.

AWS Security Audit Guidelines

You should periodically audit your security configuration to make sure it meets your current business needs. An audit gives you an opportunity to remove unneeded IAM users, roles, groups, and policies, and to make sure that your users and software have only the permissions that are required.

Following are guidelines for systematically reviewing and monitoring your AWS resources for security best practices.

Topics

When Should You Perform a Security Audit?

You should audit your security configuration in the following situations:

- On a periodic basis. You should perform the steps described in this document at regular intervals as a best practice for security.
- If there are changes in your organization, such as people leaving.
- If you have stopped using one or more individual AWS services. This is important for removing permissions that users in your account no longer need.
- If you've added or removed software in your accounts, such as applications on Amazon EC2 instances, AWS OpsWorks stacks, AWS CloudFormation templates, etc.
- If you ever suspect that an unauthorized person might have accessed your account.

General Guidelines for Auditing

As you review your account's security configuration, follow these guidelines:

- **Be thorough**. Look at all aspects of your security configuration, including those you might not use regularly.
- **Don't assume**. If you are unfamiliar with some aspect of your security configuration (for example, the reasoning behind a particular policy or the existence of a role), investigate the business need until you are satisfied.
- **Keep things simple**. To make auditing (and management) easier, use IAM groups, consistent naming schemes, and straightforward policies.

Review Your AWS Account Credentials

Take these steps when you audit your AWS account credentials:

1. If you're not using the root access keys for your account, remove them. We strongly recommend that you do not use root access keys for everyday work with AWS, and that instead you create IAM users.
2. If you do need to keep the access keys for your account, rotate them regularly.

Review Your IAM Users

Take these steps when you audit your existing IAM users:

1. Delete users that are not active.
2. Remove users from groups that they don't need to be a part of.
3. Review the policies attached to the groups the user is in. See Tips for Reviewing IAM Policies (p. 74).
4. Delete security credentials that the user doesn't need or that might have been exposed. For example, an IAM user that is used for an application does not need a password (which is necessary only to sign in to AWS websites). Similarly, if a user does not use access keys, there's no reason for the user to have one. For more information, see Managing Passwords for IAM Users and Managing Access Keys for IAM Users in the *IAM User Guide*.

You can generate and download a credential report that lists all IAM users in your account and the status of their various credentials, including passwords, access keys, and MFA devices. For passwords and access keys, the credential report shows how recently the password or access key has been used. Credentials that have not been used recently might be good candidates for removal. For more information, see Getting Credential Reports for your AWS Account in the *IAM User Guide*.

5. Rotate (change) user security credentials periodically, or immediately if you ever share them with an unauthorized person. For more information, see Managing Passwords for IAM Users and Managing Access Keys for IAM Users in the *IAM User Guide*.

Review Your IAM Groups

Take these steps when you audit your IAM groups:

1. Delete unused groups.
2. Review users in each group and remove users who don't belong. See Review Your IAM Users (p. 71) earlier.
3. Review the policies attached to the group. See Tips for Reviewing IAM Policies (p. 74).

Review Your IAM Roles

Take these steps when you audit your IAM roles:

1. Delete roles that are not in use.
2. Review the role's trust policy. Make sure that you know who the principal is and that you understand why that account or user needs to be able to assume the role.
3. Review the access policy for the role to be sure that it grants suitable permissions to whoever assumes the role—see Tips for Reviewing IAM Policies (p. 74).

Review Your IAM Providers for SAML and OpenID Connect (OIDC)

If you have created an IAM entity for establishing trust with a SAML or OIDC identity provider, take these steps:

1. Delete unused providers.
2. Download and review the AWS metadata documents for each SAML provider and make sure the documents reflect your current business needs. Alternatively, get the latest metadata documents from the SAML IdPs that you want to establish trust with and update the provider in IAM.

Review Your Mobile Apps

If you have created a mobile app that makes requests to AWS, take these steps:

1. Make sure that the mobile app does not contain embedded access keys, even if they are in encrypted storage.
2. Get temporary credentials for the app by using APIs that are designed for that purpose. We recommend that you use Amazon Cognito to manage user identity in your app. This service lets you authenticate users using Login with Amazon, Facebook, Google, or any OpenID Connect (OIDC)–compatible identity

provider. You can then use the Amazon Cognito credentials provider to manage credentials that your app uses to make requests to AWS.

If your mobile app doesn't support authentication using Login with Amazon, Facebook, Google, or any other OIDC-compatible identity provider, you can create a proxy server that can dispense temporary credentials to your app.

Review Your Amazon EC2 Security Configuration

Take the following steps for *each AWS region*:

1. Delete Amazon EC2 key pairs that are unused or that might be known to people outside your organization.
2. Review your Amazon EC2 security groups:
   - Remove security groups that no longer meet your needs.
   - Remove rules from security groups that no longer meet your needs. Make sure you know why the ports, protocols, and IP address ranges they permit have been allowed.
3. Terminate instances that aren't serving a business need or that might have been started by someone outside your organization for unapproved purposes. Remember that if an instance is started with a role, applications that run on that instance can access AWS resources using the permissions that are granted by that role.
4. Cancel spot instance requests that aren't serving a business need or that might have been made by someone outside your organization.
5. Review your Auto Scaling groups and configurations. Shut down any that no longer meet your needs or that might have been configured by someone outside your organization.

Review AWS Policies in Other Services

Review the permissions for services that use resource-based policies or that support other security mechanisms. In each case, make sure that only users and roles with a current business need have access to the service's resources, and that the permissions granted on the resources are the fewest necessary to meet your business needs.

- Review your Amazon S3 bucket policies and ACLs.
- Review your Amazon SQS queue policies.
- Review your Amazon SNS topic policies.
- Review your AWS OpsWorks permissions.
- Review your AWS KMS key policies.

Monitor Activity in Your AWS Account

Follow these guidelines for monitoring AWS activity:

- Turn on AWS CloudTrail in each account and use it in each supported region.
- Periodically examine CloudTrail log files. (CloudTrail has a number of partners who provide tools for reading and analyzing log files.)
- Enable Amazon S3 bucket logging to monitor requests made to each bucket.
- If you believe there has been unauthorized use of your account, pay particular attention to temporary credentials that have been issued. If temporary credentials have been issued that you don't recognize, disable their permissions.

- Enable billing alerts in each account and set a cost threshold that lets you know if your charges exceed your normal usage.

Tips for Reviewing IAM Policies

Policies are powerful and subtle, so it's important to study and understand the permissions that are granted by each policy. Use the following guidelines when reviewing policies:

- As a best practice, attach policies to groups instead of to individual users. If an individual user has a policy, make sure you understand why that user needs the policy.
- Make sure that IAM users, groups, and roles have only the permissions that they need.
- Use the IAM Policy Simulator to test policies that are attached to users or groups.
- Remember that a user's permissions are the result of all applicable policies—user policies, group policies, and resource-based policies (on Amazon S3 buckets, Amazon SQS queues, Amazon SNS topics, and AWS KMS keys). It's important to examine all the policies that apply to a user and to understand the complete set of permissions granted to an individual user.
- Be aware that allowing a user to create an IAM user, group, role, or policy and attach a policy to the principal entity is effectively granting that user all permissions to all resources in your account. That is, users who are allowed to create policies and attach them to a user, group, or role can grant themselves any permissions. In general, do not grant IAM permissions to users or roles whom you do not trust with full access to the resources in your account. The following list contains IAM permissions that you should review closely:
  - `iam:PutGroupPolicy`
  - `iam:PutRolePolicy`
  - `iam:PutUserPolicy`
  - `iam:CreatePolicy`
  - `iam:CreatePolicyVersion`
  - `iam:AttachGroupPolicy`
  - `iam:AttachRolePolicy`
  - `iam:AttachUserPolicy`
- Make sure policies don't grant permissions for services that you don't use. For example, if you use AWS managed policies, make sure the AWS managed policies that are in use in your account are for services that you actually use. To find out which AWS managed policies are in use in your account, use the IAM `GetAccountAuthorizationDetails` API (AWS CLI command: `aws iam get-account-authorization-details`).
- If the policy grants a user permission to launch an Amazon EC2 instance, it might also allow the `iam:PassRole` action, but if so it should explicitly list the roles that the user is allowed to pass to the Amazon EC2 instance.
- Closely examine any values for the `Action` or `Resource` element that include `*`. It's a best practice to grant `Allow` access to only the individual actions and resources that users need. However, the following are reasons that it might be suitable to use `*` in a policy:
  - The policy is designed to grant administrative-level privileges.
  - The wildcard character is used for a set of similar actions (for example, `Describe*`) as a convenience, and you are comfortable with the complete list of actions that are referenced in this way.
  - The wildcard character is used to indicate a class of resources or a resource path (e.g., `arn:aws:iam::`*`account-id`*`:users/division_abc/*`), and you are comfortable granting access to all of the resources in that class or path.
  - A service action does not support resource-level permissions, and the only choice for a resource is `*`.
- Examine policy names to make sure they reflect the policy's function. For example, although a policy might have a name that includes "read only," the policy might actually grant write or change permissions.

More Information

For information about managing IAM resources, see the following:

- IAM Users and Groups in the *IAM User Guide*.
- Permissions and Policies in the *IAM User Guide*.
- IAM Roles (Delegation and Federation) in the *IAM User Guide*.
- IAM Policy Simulator in the *Using IAM Policy Simulator* guide.

For more information about Amazon EC2 security, see the following:

- Network and Security in the *Amazon EC2 User Guide for Linux Instances*.
- Demystifying EC2 Resource-Level Permissions on the AWS Security Blog.

For more information about monitoring an AWS account, see the re:Invent 2013 presentation "Intrusion Detection in the Cloud" (video, PDF of slide presentation). You can also download a sample Python program that shows how to automate security auditing functions.

Amazon Resource Names (ARNs) and AWS Service Namespaces

Amazon Resource Names (ARNs) uniquely identify AWS resources. We require an ARN when you need to specify a resource unambiguously across all of AWS, such as in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls.

Topics

ARN Format

Here are some example ARNs:

```
<!-- Elastic Beanstalk application version -->
arn:aws:elasticbeanstalk:us-east-1:123456789012:environment/My App/MyEnvironment

<!-- IAM user name -->
arn:aws:iam::123456789012:user/David

<!-- Amazon RDS instance used for tagging -->
arn:aws:rds:eu-west-1:123456789012:db:mysql-db

<!-- Object in an Amazon S3 bucket -->
arn:aws:s3:::my_corporate_bucket/exampleobject.png
```

The following are the general formats for ARNs; the specific components and values used depend on the AWS service.

```
arn:partition:service:region:account-id:resource
arn:partition:service:region:account-id:resourcetype/resource
```

```
arn:partition:service:region:account-id:resourcetype:resource
```

partition

The partition that the resource is in. For standard AWS regions, the partition is `aws`. If you have resources in other partitions, the partition is `aws-`*partitionname*. For example, the partition for resources in the China (Beijing) region is `aws-cn`.

service

The service namespace that identifies the AWS product (for example, Amazon S3, IAM, or Amazon RDS). For a list of namespaces, see AWS Service Namespaces (p. 94).

region

The region the resource resides in. Note that the ARNs for some resources do not require a region, so this component might be omitted.

account

The ID (p. 66) of the AWS account that owns the resource, without the hyphens. For example, 123456789012. Note that the ARNs for some resources don't require an account number, so this component might be omitted.

resource, *resourcetype*:*resource*, or *resourcetype*/*resource*

The content of this part of the ARN varies by service. It often includes an indicator of the type of resource—for example, an IAM user or Amazon RDS database —followed by a slash (`/`) or a colon (`:`), followed by the resource name itself. Some services allows paths for resource names, as described in Paths in ARNs (p. 93).

Example ARNs

The following sections provide syntax and examples of the ARNs for different services. For more information about using ARNs in a specific AWS service, see the documentation for that service.

Some services support IAM resource-level permissions. For more information, see AWS Services That Work with IAM.

Topics

Amazon API Gateway

Syntax:

```
arn:aws:apigateway:region::resource-path
```

```
arn:aws:execute-api:region:account-id:api-id/stage-name/HTTP-VERB/resource-path
```

Examples:

```
arn:aws:apigateway:us-east-1::/restapis/a123456789012bc3de45678901f23a45/*
arn:aws:apigateway:us-east-1::a123456789012bc3de45678901f23a45:/test/mydemoresource/*
arn:aws:apigateway:*::a123456789012bc3de45678901f23a45:/*/petstorewalkthrough/pets
arn:aws:execute-api:us-east-1:123456789012:qsxrty/test/GET/mydemoresource/*
```

AWS Artifact

Syntax:

```
arn:aws:artifact:::report-package/document-type/report-type
```

Examples:

```
arn:aws:artifact:::report-package/Certifications and Attestations/SOC/*
arn:aws:artifact:::report-package/Certifications and Attestations/ISO/*
arn:aws:artifact:::report-package/Certifications and Attestations/PCI/*
```

Auto Scaling

Syntax:

```
arn:aws:autoscaling:region:account-
id:scalingPolicy:policyid:autoScalingGroupName/groupfriendlyname:policyname/
policyfriendlyname
arn:aws:autoscaling:region:account-
id:autoScalingGroup:groupid:autoScalingGroupName/groupfriendlyname
```

Example:

```
arn:aws:autoscaling:us-east-1:123456789012:scalingPolicy:c7a27f55-d35e-4153-
b044-8ca9155fc467:autoScalingGroupName/my-test-asg1:policyName/my-scaleout-policy
```

AWS Certificate Manager

Syntax:

```
arn:aws:acm:region:account-id:certificate/certificate-id
```

Example:

```
arn:aws:acm:us-east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012
```

AWS CloudFormation

Syntax:

```
arn:aws:cloudformation:region:account-id:stack/stackname/additionalidentifier
```

```
arn:aws:cloudformation:region:account-id:changeSet/changesetname/additionalidentifier
```

Examples:

```
arn:aws:cloudformation:us-east-1:123456789012:stack/MyProductionStack/abc9dbf0-43c2-11e3-
a6e8-50fa526be49c
```

```
arn:aws:cloudformation:us-east-1:123456789012:changeSet/MyProductionChangeSet/
abc9dbf0-43c2-11e3-a6e8-50fa526be49c
```

Amazon CloudSearch

Syntax:

```
arn:aws:cloudsearch:region:account-id:domain/domainname
```

Example:

```
arn:aws:cloudsearch:us-east-1:123456789012:domain/imdb-movies
```

AWS CloudTrail

Syntax:

```
arn:aws:cloudtrail:region:account-id:trail/trailname
```

Example:

```
arn:aws:cloudtrail:us-east-1:123456789012:trail/mytrailname
```

Amazon CloudWatch Events

Syntax:

```
arn:aws:events:region:*:*
```

Examples:

```
arn:aws:events:us-east-1:*:*
arn:aws:events:us-east-1:123456789012:*
arn:aws:events:us-east-1:123456789012:rule/my-rule
```

Amazon CloudWatch Logs

Syntax:

```
arn:aws:logs:region:*:*
```

Examples:

```
arn:aws:logs:us-east-1:*:*
arn:aws:logs:us-east-1:123456789012:*
arn:aws:logs:us-east-1:123456789012:log-group:my-log-group
arn:aws:logs:us-east-1:123456789012:log-group:my-log-group:*
arn:aws:logs:us-east-1:123456789012:log-group:my-log-group*
arn:aws:logs:us-east-1:123456789012:log-group:my-log-group:log-stream:my-log-stream
arn:aws:logs:us-east-1:123456789012:log-group:my-log-group:log-stream:my-log-stream*
arn:aws:logs:us-east-1:123456789012:log-group:my-log-group*:log-stream:my-log-stream*
```

AWS CodeBuild

Syntax:

```
arn:aws:codebuild:region:account-id:resourcetype/resource
```

Examples:

```
arn:aws:codebuild:us-east-1:123456789012:project/my-demo-project
arn:aws:codebuild:us-east-1:123456789012:build/my-demo-
project:7b7416ae-89b4-46cc-8236-61129df660ad
```

AWS CodeCommit

Syntax:

```
arn:aws:codecommit:region:account-id:resource-specifier
```

Example:

```
arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo
```

AWS CodeDeploy

Syntax:

```
arn:aws:codedeploy:region:account-id:resource-type:resource-specifier
arn:aws:codedeploy:region:account-id:resource-type/resource-specifier
```

Example:

```
arn:aws:codedeploy:us-east-1:123456789012:application:WordPress_App
arn:aws:codedeploy:us-east-1:123456789012:instance/AssetTag*
```

Amazon Cognito Your User Pools

Syntax:

```
arn:aws:cognito-idp:region:account-id:userpool/user-pool-id
```

Example:

```
arn:aws:cognito-idp:us-east-1:123456789012:userpool/us-east-1:1a1a1a1a-
ffff-1111-9999-12345678
```

Amazon Cognito Federated Identities

Syntax:

```
arn:aws:cognito-identity:region:account-id:identitypool/identity-pool-id
```

Example:

```
arn:aws:cognito-identity:us-east-1:123456789012:/identitypool/us-east-1:1a1a1a1a-
ffff-1111-9999-12345678
```

Amazon Cognito Sync

Syntax:

```
arn:aws:cognito-sync:region:account-id:identitypool/identity-pool-id
```

```
arn:aws:cognito-sync:region:account-id:identitypool/identity-pool-id/identity/identity-id
```

```
arn:aws:cognito-sync:region:account-id:identitypool/identity-pool-id/identity/identity-id/
dataset/dataset-name
```

Example:

```
arn:aws:cognito-sync:us-east-1:123456789012:identitypool/us-east-1:1a1a1a1a-
ffff-1111-9999-12345678
```

AWS Config

Syntax:

```
arn:aws:config:region:account-id:config-rule/config-rule-name
```

Example:

```
arn:aws:config:us-east-1:123456789012:config-rule/MyConfigRule
```

AWS CodePipeline

Syntax:

```
arn:aws:codepipeline:region:account-id:resource-specifier
```

Example:

```
arn:aws:codepipeline:us-east-1:123456789012:MyDemoPipeline
```

AWS Direct Connect

Syntax:

```
arn:aws:directconnect:region:account-id:dxcon/connection-id
arn:aws:directconnect:region:account-id:dxlag/lag-id
arn:aws:directconnect:region:account-id:dxvif/virtual-interface-id
```

Examples:

```
arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-fgase048
arn:aws:directconnect:us-east-1:123456789012:dxlag/dxlag-ffy7zraq
arn:aws:directconnect:us-east-1:123456789012:dxvif/dxvif-fgrb110x
```

Amazon DynamoDB

Syntax:

```
arn:aws:dynamodb:region:account-id:table/tablename
```

Example:

```
arn:aws:dynamodb:us-east-1:123456789012:table/books_table
```

Amazon EC2 Container Registry (Amazon ECR)

Syntax:

```
arn:aws:ecr:region:account-id:repository/repository-name
```

Example:

```
arn:aws:ecr:us-east-1:123456789012:repository/my-repository
```

Amazon EC2 Container Service (Amazon ECS)

Syntax:

```
arn:aws:ecs:region:account-id:cluster/cluster-name
arn:aws:ecs:region:account-id:container-instance/container-instance-id
arn:aws:ecs:region:account-id:task-definition/task-definition-family-name:task-definition-
revision-number
arn:aws:ecs:region:account-id:service/service-name
arn:aws:ecs:region:account-id:task/task-id
arn:aws:ecs:region:account-id:container/container-id
```

Examples:

```
arn:aws:ecs:us-east-1:123456789012:cluster/my-cluster
arn:aws:ecs:us-east-1:123456789012:container-instance/403125b0-555c-4473-86b5-65982db28a6d
arn:aws:ecs:us-east-1:123456789012:task-definition/hello_world:8
arn:aws:ecs:us-east-1:123456789012:service/sample-webapp
arn:aws:ecs:us-east-1:123456789012:task/1abf0f6d-a411-4033-b8eb-a4eed3ad252a
```

```
arn:aws:ecs:us-east-1:123456789012:container/476e7c41-17f2-4c17-9d14-412566202c8a
```

Amazon Elastic Compute Cloud (Amazon EC2)

Syntax:

```
arn:aws:ec2:region:account-id:customer-gateway/cgw-id
arn:aws:ec2:region:account_id:dedicated-host/host_id
arn:aws:ec2:region:account-id:dhcp-options/dhcp-options-id
arn:aws:ec2:region::image/image-id
arn:aws:ec2:region:account-id:instance/instance-id
arn:aws:iam::account:instance-profile/instance-profile-name
arn:aws:ec2:region:account-id:internet-gateway/igw-id
arn:aws:ec2:region:account-id:key-pair/key-pair-name
arn:aws:ec2:region:account-id:network-acl/nacl-id
arn:aws:ec2:region:account-id:network-interface/eni-id
arn:aws:ec2:region:account-id:placement-group/placement-group-name
arn:aws:ec2:region:account-id:route-table/route-table-id
arn:aws:ec2:region:account-id:security-group/security-group-id
arn:aws:ec2:region:account-id:snapshot/snapshot-id
arn:aws:ec2:region:account-id:subnet/subnet-id
arn:aws:ec2:region:account-id:volume/volume-id
arn:aws:ec2:region:account-id:vpc/vpc-id
arn:aws:ec2:region:account-id:vpc-peering-connection/vpc-peering-connection-id
arn:aws:ec2:region:account-id:vpn-connection/vpn-id
arn:aws:ec2:region:account-id:vpn-gateway/vgw-id
```

Examples:

```
arn:aws:ec2:us-east-1:123456789012:dedicated-host/h-12345678
arn:aws:ec2:us-east-1::image/ami-1a2b3c4d
arn:aws:ec2:us-east-1:123456789012:instance/*
arn:aws:ec2:us-east-1:123456789012:volume/*
arn:aws:ec2:us-east-1:123456789012:volume/vol-1a2b3c4d
```

AWS Elastic Beanstalk

Syntax:

```
arn:aws:elasticbeanstalk:region:account-id:application/applicationname
arn:aws:elasticbeanstalk:region:account-id:applicationversion/applicationname/versionlabel
arn:aws:elasticbeanstalk:region:account-id:environment/applicationname/environmentname
arn:aws:elasticbeanstalk:region::solutionstack/solutionstackname
arn:aws:elasticbeanstalk:region:account-
id:configurationtemplate/applicationname/templatename
```

Examples:

```
arn:aws:elasticbeanstalk:us-east-1:123456789012:application/My App
arn:aws:elasticbeanstalk:us-east-1:123456789012:applicationversion/My App/My Version
arn:aws:elasticbeanstalk:us-east-1:123456789012:environment/My App/MyEnvironment
arn:aws:elasticbeanstalk:us-east-1::solutionstack/32bit Amazon Linux running Tomcat 7
arn:aws:elasticbeanstalk:us-east-1:123456789012:configurationtemplate/My App/My Template
```

Amazon Elastic File System

Syntax:

```
arn:aws:elasticfilesystem:region:account-id:file-system/file-system-id
```

Example:

```
arn:aws:elasticfilesystem:us-east-1:123456789012:file-system-id/fs12345678
```

Elastic Load Balancing (Application Load Balancer)

Syntax:

```
arn:aws:elasticloadbalancing:region:account-id:loadbalancer/app/load-balancer-name/load-balancer-id
arn:aws:elasticloadbalancing:region:account-id:listener/app/load-balancer-name/load-balancer-id/listener-id
arn:aws:elasticloadbalancing:region:account-id:listener-rule/app/load-balancer-name/load-balancer-id/listener-id/rule-id
arn:aws:elasticloadbalancing:region:account-id:targetgroup/target-group-name/target-group-id
```

Examples:

```
arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
arn:aws:elasticloadbalancing:us-east-1:123456789012:listener-rule/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

Elastic Load Balancing (Classic Load Balancer)

Syntax:

```
arn:aws:elasticloadbalancing:region:account-id:loadbalancer/name
```

Example:

```
arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/my-load-balancer
```

Amazon Elastic Transcoder

Syntax:

```
arn:aws:elastictranscoder:region:account-id:resource/id
```

Example:

```
arn:aws:elastictranscoder:us-east-1:123456789012:preset/*
```

Amazon ElastiCache

Syntax:

```
arn:aws:elasticache:region:account-id:resourcetype:resourcename
```

Examples:

```
arn:aws:elasticache:us-west-2:123456789012:cluster:myCluster
arn:aws:elasticache:us-west-2:123456789012:snapshot:mySnapshot
```

Amazon Elasticsearch Service

Syntax:

```
arn:aws:es:region:account-id:domain/domain-name
```

Example:

```
arn:aws:es:us-east-1:123456789012:domain/streaming-logs
```

Amazon Glacier

Syntax:

```
arn:aws:glacier:region:account-id:vaults/vaultname
```

Examples:

```
arn:aws:glacier:us-east-1:123456789012:vaults/examplevault
arn:aws:glacier:us-east-1:123456789012:vaults/example*
arn:aws:glacier:us-east-1:123456789012:vaults/*
```

AWS Health / Personal Health Dashboard

Syntax:

```
arn:aws:health:region::event/event-id
arn:aws:health:region:account-id:entity/entity-id
```

Examples:

```
arn:aws:health:us-east-1::event/AWS_EC2_EXAMPLE_ID
arn:aws:health:us-east-1:123456789012:entity/AVh5GGT7ul1arKr1sE1K
```

AWS Identity and Access Management (IAM)

Syntax:

```
arn:aws:iam::account-id:root
arn:aws:iam::account-id:user/user-name
arn:aws:iam::account-id:group/group-name
arn:aws:iam::account-id:role/role-name
arn:aws:iam::account-id:policy/policy-name
arn:aws:iam::account-id:instance-profile/instance-profile-name
```

```
arn:aws:sts::account-id:federated-user/user-name
arn:aws:sts::account-id:assumed-role/role-name/role-session-name
arn:aws:iam::account-id:mfa/virtual-device-name
arn:aws:iam::account-id:server-certificate/certificate-name
arn:aws:iam::account-id:saml-provider/provider-name
arn:aws:iam::account-id:oidc-provider/provider-name
```

Examples:

```
arn:aws:iam::123456789012:root
arn:aws:iam::123456789012:user/Bob
arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Bob
arn:aws:iam::123456789012:group/Developers
arn:aws:iam::123456789012:group/division_abc/subdivision_xyz/product_A/Developers
arn:aws:iam::123456789012:role/S3Access
arn:aws:iam::123456789012:role/application_abc/component_xyz/S3Access
arn:aws:iam::123456789012:policy/UsersManageOwnCredentials
arn:aws:iam::123456789012:policy/division_abc/subdivision_xyz/UsersManageOwnCredentials
arn:aws:iam::123456789012:instance-profile/Webserver
arn:aws:sts::123456789012:federated-user/Bob
arn:aws:sts::123456789012:assumed-role/Accounting-Role/Mary
arn:aws:iam::123456789012:mfa/BobJonesMFA
arn:aws:iam::123456789012:server-certificate/ProdServerCert
arn:aws:iam::123456789012:server-certificate/division_abc/subdivision_xyz/ProdServerCert
arn:aws:iam::123456789012:saml-provider/ADFSProvider
arn:aws:iam::123456789012:oidc-provider/GoogleProvider
```

For more information about IAM ARNs, see IAM ARNs in *IAM User Guide*.

AWS IoT

Syntax:

```
arn:aws:iot:your-region:account-id:cert/cert-ID
arn:aws:iot:your-region:account-id:policy/policy-name
arn:aws:iot:your-region:account-id:rule/rule-name
arn:aws:iot:your-region:account-id:client/client-id/rule-name
```

Examples:

```
arn:aws:iot:your-
region:123456789012:cert/123a456b789c123d456e789f123a456b789c123d456e789f123a456b789c123c456d7
arn:aws:iot:123456789012:policy/MyIoTPolicy
arn:aws:iot:your-region:123456789012:rule/MyIoTRule
arn:aws:iot:your-region:123456789012:client/client101
```

AWS Key Management Service (AWS KMS)

Syntax:

```
arn:aws:kms:region:account-id:key/key-id
arn:aws:kms:region:account-id:alias/alias
```

Examples:

```
arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012
arn:aws:kms:us-east-1:123456789012:alias/example-alias
```

Amazon Kinesis Firehose (Firehose)

Syntax:

```
arn:aws:firehose:region:account-id:deliverystream/delivery-stream-name
```

Example:

```
arn:aws:firehose:us-east-1:123456789012:deliverystream/example-stream-name
```

Amazon Kinesis Streams (Streams)

Syntax:

```
arn:aws:kinesis:region:account-id:stream/stream-name
```

Example:

```
arn:aws:kinesis:us-east-1:123456789012:stream/example-stream-name
```

AWS Lambda (Lambda)

Syntax:

```
arn:aws:lambda:region:account-id:function:function-name
arn:aws:lambda:region:account-id:function:function-name:alias-name
arn:aws:lambda:region:account-id:function:function-name:version
arn:aws:lambda:region:account-id:event-source-mappings:event-source-mapping-id
```

Examples:

```
arn:aws:lambda:us-east-1:123456789012:function:ProcessKinesisRecords
arn:aws:lambda:us-east-1:123456789012:function:ProcessKinesisRecords:your alias
arn:aws:lambda:us-east-1:123456789012:function:ProcessKinesisRecords:1.0
arn:aws:lambda:us-east-1:123456789012:event-source-mappings:kinesis-stream-arn
```

Amazon Machine Learning (Amazon ML)

Syntax:

```
arn:aws:machinelearning:region:account-id:datasource/datasourceID
arn:aws:machinelearning:region:account-id:mlmodel/mlmodelID
arn:aws:machinelearning:region:account-id:batchprediction/batchpredictionID
arn:aws:machinelearning:region:account-id:evaluation/evaluationID
```

Examples:

```
arn:aws:machinelearning:us-east-1:123456789012:datasource/my-datasource-1
arn:aws:machinelearning:us-east-1:123456789012:mlmodel/my-mlmodel
arn:aws:machinelearning:us-east-1:123456789012:batchprediction/my-batchprediction
arn:aws:machinelearning:us-east-1:123456789012:evaluation/my-evaluation
```

AWS Organizations

Syntax:

```
arn:aws:organizations:region:master-account-id:organization/o-organization-id
arn:aws:organizations:region:master-account-id:root/o-organization-id/r-root-id
arn:aws:organizations:region:master-account-id:account/o-organization-id/account-id
arn:aws:organizations:region:master-account-id:ou/o-organization-id/r-root-id/
ou-organizational-unit-id
arn:aws:organizations:region:master-account-id:policy/o-organization-id/policy-type/
p-policy-id
arn:aws:organizations:region:master-account-id:handshake/o-organization-id/handshake-type/
h-handshake-id
```

Example:

```
arn:aws:organizations:us-east-1:123456789012:organization/o-a1b2c3d4e5example
arn:aws:organizations:us-east-1:123456789012:root/o-a1b2c3d4e5/r-f6g7h8i9j0example
arn:aws:organizations:us-east-1:123456789012:account/o-a1b2c3d4e5/123456789012
arn:aws:organizations:us-east-1:123456789012:ou/o-a1b2c3d4e5/ou-1a2b3c-k9l8m7n6o5example
arn:aws:organizations:us-east-1:123456789012:policy/o-a1b2c3d4e5/service_control_policy/p-
p4q3r2s1t0example
arn:aws:organizations:us-east-1:123456789012:handshake/o-a1b2c3d4e5/h-u2v4w5x8y0example
```

AWS Mobile Hub

Syntax:

```
arn:aws:mobilehub:region:account-id:project/projectID
```

Examples:

```
arn:aws:mobilehub:us-east-1:123456789012:project/a01234567-b012345678-123c-d013456789abc
```

Amazon Polly

Syntax:

```
arn:aws:polly:region:account-id:lexicon/LexiconName
```

Example:

```
arn:aws:polly:us-east-1:123456789012:lexicon/myLexicon
```

Amazon Redshift

Syntax:

```
arn:aws:redshift:region:account-id:cluster:clustername
arn:aws:redshift:region:account-id:dbuser:clustername/dbusername
arn:aws:redshift:region:account-id:parametergroup:parametergroupname
arn:aws:redshift:region:account-id:securitygroup:securitygroupname
arn:aws:redshift:region:account-id:snapshot:clustername/snapshotname
```

```
arn:aws:redshift:region:account-id:subnetgroup:subnetgroupname
```

Examples:

```
arn:aws:redshift:us-east-1:123456789012:cluster:my-cluster
arn:aws:redshift:us-east-1:123456789012:my-cluster/my-dbuser-name
arn:aws:redshift:us-east-1:123456789012:parametergroup:my-parameter-group
arn:aws:redshift:us-east-1:123456789012:securitygroup:my-public-group
arn:aws:redshift:us-east-1:123456789012:snapshot:my-cluster/my-snapshot20130807
arn:aws:redshift:us-east-1:123456789012:subnetgroup:my-subnet-10
```

Amazon Relational Database Service (Amazon RDS)

ARNs are used in Amazon RDS only with tags for DB instances. For more information, see Tagging a DB Instance in the *Amazon Relational Database Service User Guide*.

Syntax:

```
arn:aws:rds:region:account-id:db:db-instance-name
arn:aws:rds:region:account-id:snapshot:snapshot-name
arn:aws:rds:region:account-id:cluster:db-cluster-name
arn:aws:rds:region:account-id:cluster-snapshot:cluster-snapshot-name
arn:aws:rds:region:account-id:og:option-group-name
arn:aws:rds:region:account-id:pg:parameter-group-name
arn:aws:rds:region:account-id:cluster-pg:cluster-parameter-group-name
arn:aws:rds:region:account-id:secgrp:security-group-name
arn:aws:rds:region:account-id:subgrp:subnet-group-name
arn:aws:rds:region:account-id:es:subscription-name
```

Examples:

```
arn:aws:rds:us-east-1:123456789012:db:mysql-db-instance1
arn:aws:rds:us-east-1:123456789012:snapshot:my-snapshot2
arn:aws:rds:us-east-1:123456789012:cluster:my-cluster1
arn:aws:rds:us-east-1:123456789012:cluster-snapshot:cluster1-snapshot7
arn:aws:rds:us-east-1:123456789012:og:mysql-option-group1
arn:aws:rds:us-east-1:123456789012:pg:mysql-repl-pg1
arn:aws:rds:us-east-1:123456789012:cluster-pg:aurora-pg3
arn:aws:rds:us-east-1:123456789012:secgrp:dev-secgrp2
arn:aws:rds:us-east-1:123456789012:subgrp:prod-subgrp1
arn:aws:rds:us-east-1:123456789012:es:monitor-events2
```

Amazon Route 53

Syntax:

```
arn:aws:route53:::hostedzone/zoneid
arn:aws:route53:::change/changeid
```

Note that Amazon Route 53 does not require an account number or region in ARNs.

Examples:

```
arn:aws:route53:::hostedzone/Z148QEXAMPLE8V
arn:aws:route53:::change/C2RDJ5EXAMPLE2
arn:aws:route53:::change/*
```

Amazon EC2 Systems Manager (SSM)

Syntax:

```
arn:aws:ssm:region:account-id:document/document_name
arn:aws:ssm:region:account-id:parameter/parameter_name
arn:aws:ssm:region:account-id:patchbaseline/baseline_id
arn:aws:ssm:region:account-id:maintenancewindow/window_id
arn:aws:ssm:region:account-id:automation-execution/execution_id
arn:aws:ssm:region:account-id:automation-Activity/activity_name
arn:aws:ssm:region:account-id:automation-definition/definitionName:version
arn:aws:ssm:region:account-id:managed-instance/instance_id
arn:aws:ssm:region:account-id:managed-instance-inventory/instance_id
```

Examples:

```
arn:aws:ssm:us-east-1:123456789012:document/highAvailabilityServerSetup
arn:aws:ssm:us-east-1:123456789012:parameter/myParameterName
arn:aws:ssm:us-east-1:123456789012:patchbaseline/pb-12345678901234567
arn:aws:ssm:us-east-1:123456789012:maintenancewindow/mw-12345678901234567
arn:aws:ssm:us-east-1:123456789012:automation-execution/123456-6789-1a2b3-c4d5-e1a2b3c4d
arn:aws:ssm:us-east-1:123456789012:automation-activity/myActivityName
arn:aws:ssm:us-east-1:123456789012:automation-definition/myDefinitionName:1.0
arn:aws:ssm:us-east-1:123456789012:managed-instance/mi-12345678901234567
arn:aws:ssm:us-east-1:123456789012:managed-instance-inventory/i-12345661
```

Amazon Simple Notification Service (Amazon SNS)

Syntax:

```
arn:aws:sns:region:account-id:topicname
arn:aws:sns:region:account-id:topicname:subscriptionid
```

Examples:

```
arn:aws:sns:*:123456789012:my_corporate_topic
arn:aws:sns:us-east-1:123456789012:my_corporate_topic:02034b43-fefa-4e07-a5eb-3be56f8c54ce
```

Amazon Simple Queue Service (Amazon SQS)

Syntax:

```
arn:aws:sqs:region:account-id:queuename
```

Example:

```
arn:aws:sqs:us-east-1:123456789012:queue1
```

Amazon Simple Storage Service (Amazon S3)

Syntax:

```
arn:aws:s3:::bucket_name
arn:aws:s3:::bucket_name/key_name
```

Note

Amazon S3 does not require an account number or region in ARNs. If you specify an ARN for a policy, you can also use a wildcard "*" character in the relative-ID part of the ARN.

Examples:

```
arn:aws:s3:::my_corporate_bucket
arn:aws:s3:::my_corporate_bucket/exampleobject.png
arn:aws:s3:::my_corporate_bucket/*
arn:aws:s3:::my_corporate_bucket/Development/*
```

For more information, see Specifying Resources in a Policy in the *Amazon Simple Storage Service Developer Guide*.

Amazon Simple Workflow Service (Amazon SWF)

Syntax:

```
arn:aws:swf:region:account-id:/domain/domain_name
```

Examples:

```
arn:aws:swf:us-east-1:123456789012:/domain/department1
arn:aws:swf:*:123456789012:/domain/*
```

AWS Step Functions

Syntax:

```
arn:aws:states:region:account-id:activity:activityName
arn:aws:states:region:account-id:stateMachine:stateMachineName
arn:aws:states:region:account-id:execution:stateMachineName:executionName
```

Examples:

```
arn:aws:states:us-east-1:123456789012:activity:HelloActivity
arn:aws:states:us-east-1:123456789012:stateMachine:HelloStateMachine
arn:aws:states:us-
east-1:123456789012:execution:HelloStateMachine:HelloStateMachineExecution
```

AWS Storage Gateway

Syntax:

```
arn:aws:storagegateway:region:account-id:gateway/gateway-id
arn:aws:storagegateway:region:account-id:gateway/gateway-id/volume/volume-id
arn:aws:storagegateway:region:account-id:tape/tapebarcode
arn:aws:storagegateway:region:account-id:gateway/gateway-id/target/iSCSItarget
arn:aws:storagegateway:region:account-id:gateway/gateway-id/device/vtldevice
```

Examples:

```
arn:aws:storagegateway:us-east-1:123456789012:gateway/sgw-12A3456B
arn:aws:storagegateway:us-east-1:123456789012:gateway/sgw-12A3456B/volume/vol-1122AABB
arn:aws:storagegateway:us-east-1:123456789012:tape/AMZNC8A26D
```

```
arn:aws:storagegateway:us-east-1:123456789012:gateway/sgw-12A3456B/target/
iqn.1997-05.com.amazon:vol-1122AABB
arn:aws:storagegateway:us-east-1:123456789012:gateway/sgw-12A3456B/device/AMZN_SGW-
FF22CCDD_TAPEDRIVE_00010
```

Note
For each AWS Storage Gateway resource, you can specify a wild card (*).

AWS Trusted Advisor

Syntax:

```
arn:aws:trustedadvisor:*:account-id:checks/categorycode/checkid
```

Example:

```
arn:aws:trustedadvisor:*:123456789012:checks/fault_tolerance/BueAdJ7NrP
```

AWS WAF

Syntax:

```
arn:aws:waf::account-id:resource-type/resource-id
```

Examples:

```
arn:aws:waf::123456789012:rule/41b5b052-1e4a-426b-8149-3595be6342c2
arn:aws:waf::123456789012:webacl/3bffd3ed-fa2e-445e-869f-a6a7cf153fd3
arn:aws:waf::123456789012:ipset/3f74bd8c-f046-4970-a1a7-41aa52e05480
arn:aws:waf::123456789012:bytematchset/d131bc0b-57be-4536-af1d-4894fd28acc4
arn:aws:waf::123456789012:sqlinjectionset/2be79d6f-2f41-4c9b-8192-d719676873f0
arn:aws:waf::123456789012:changetoken/03ba2197-fc98-4ac0-a67d-5b839762b16b
```

Paths in ARNs

Some services let you specify a path for the resource name. For example, in Amazon S3, the resource identifier is an object name that can include slashes (`/`) to form a path. Similarly, IAM user names and group names can include paths.

In some circumstances, paths can include a wildcard character, namely an asterisk (`*`). For example, if you are writing an IAM policy and in the `Resource` element you want to specify all IAM users that have the path `product_1234`, you can use a wildcard like this:

```
arn:aws:iam::123456789012:user/Development/product_1234/*
```

Similarly, in the `Resource` element of an IAM policy, at the end of the ARN you can specify `user/*` to mean all users or `group/*` to mean all groups, as in the following examples:

```
"Resource":"arn:aws:iam::123456789012:user/*"
"Resource":"arn:aws:iam::123456789012:group/*"
```

You cannot use a wildcard to specify all users in the `Principal` element in a resource-based policy or a role trust policy. Groups are not supported as principals in any policy.

The following example shows ARNs for an Amazon S3 bucket in which the resource name includes a path:

```
arn:aws:s3:::my_corporate_bucket/*
arn:aws:s3:::my_corporate_bucket/Development/*
```

You cannot use a wildcard in the portion of the ARN that specifies the resource type, such as the term `user` in an IAM ARN.

The following is not allowed:

```
arn:aws:iam::123456789012:u*
```

AWS Service Namespaces

When you create AWS IAM policies or work with Amazon Resource Names (ARNs), you identify an AWS service using a *namespace*. For example, the namespace for Amazon S3 is `s3`, and the namespace for Amazon EC2 is `ec2`. You use namespaces when identifying actions and resources.

The following example shows an IAM policy where the value of the `Action` elements and the values in the `Resource` and `Condition` elements use namespaces to identify the services for the actions and resources.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:*",
      "Resource": [
        "arn:aws:ec2:us-west-2:123456789012:customer-gateway/*",
        "arn:aws:ec2:us-west-2:123456789012:dhcp-options/*",
        "arn:aws:ec2:us-west-2::image/*",
        "arn:aws:ec2:us-west-2:123456789012:instance/*",
        "arn:aws:iam::123456789012:instance-profile/*",
        "arn:aws:ec2:us-west-2:123456789012:internet-gateway/*",
        "arn:aws:ec2:us-west-2:123456789012:key-pair/*",
        "arn:aws:ec2:us-west-2:123456789012:network-acl/*",
        "arn:aws:ec2:us-west-2:123456789012:network-interface/*",
        "arn:aws:ec2:us-west-2:123456789012:placement-group/*",
        "arn:aws:ec2:us-west-2:123456789012:route-table/*",
        "arn:aws:ec2:us-west-2:123456789012:security-group/*",
        "arn:aws:ec2:us-west-2::snapshot/*",
        "arn:aws:ec2:us-west-2:123456789012:subnet/*",
        "arn:aws:ec2:us-west-2:123456789012:volume/*",
        "arn:aws:ec2:us-west-2:123456789012:vpc/*",
        "arn:aws:ec2:us-west-2:123456789012:vpc-peering-connection/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::example_bucket/marketing/*"
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket*",
      "Resource": "arn:aws:s3:::example_bucket",
      "Condition": {"StringLike": {"s3:prefix": "marketing/*"}}
    }
  ]
}
```

The following table contains the namespace for each AWS service.

Service	Namespace
API Gateway	`apigateway`
Amazon AppStream	`appstream`
AWS Artifact	`artifact`
Auto Scaling	`autoscaling`
AWS Billing and Cost Management	`aws-portal`
AWS Certificate Manager (ACM)	`acm`
AWS CloudFormation	`cloudformation`
Amazon CloudFront	`cloudfront`
AWS CloudHSM	`cloudhsm`
Amazon CloudSearch	`cloudsearch`
AWS CloudTrail	`cloudtrail`
Amazon CloudWatch	`cloudwatch`
Amazon CloudWatch Events	`events`
Amazon CloudWatch Logs	`logs`
AWS CodeBuild	`codebuild`
AWS CodeCommit	`codecommit`
AWS CodeDeploy	`codedeploy`
AWS CodePipeline	`codepipeline`
Amazon Cognito Your User Pools	`cognito-idp`
Amazon Cognito Federated Identities	`cognito-identity`
Amazon Cognito Sync	`cognito-sync`
AWS Config	`config`
AWS Data Pipeline	`datapipeline`
AWS Database Migration Service (AWS DMS)	`dms`
AWS Device Farm	`devicefarm`
AWS Direct Connect	`directconnect`
AWS Directory Service	`ds`
Amazon DynamoDB	`dynamodb`
Amazon Elastic Compute Cloud (Amazon EC2)	`ec2`
Amazon EC2 Container Registry (Amazon ECR)	`ecr`

Service	Namespace
Amazon EC2 Container Service (Amazon ECS)	`ecs`
Amazon EC2 Systems Manager (SSM)	`ssm`
AWS Elastic Beanstalk	`elasticbeanstalk`
Amazon Elastic File System (Amazon EFS)	`elasticfilesystem`
Elastic Load Balancing	`elasticloadbalancing`
Amazon EMR	`elasticmapreduce`
Amazon Elastic Transcoder	`elastictranscoder`
Amazon ElastiCache	`elasticache`
Amazon Elasticsearch Service (Amazon ES)	`es`
Amazon GameLift	`gamelift`
Amazon Glacier	`glacier`
AWS Health / Personal Health Dashboard	`health`
AWS Identity and Access Management (IAM)	`iam`
AWS Import/Export	`importexport`
Amazon Inspector	`inspector`
AWS IoT	`iot`
AWS Key Management Service (AWS KMS)	`kms`
Amazon Kinesis Analytics	`kinesisanalytics`
Amazon Kinesis Firehose	`firehose`
Amazon Kinesis Streams	`kinesis`
AWS Lambda	`lambda`
Amazon Lightsail	`lightsail`
Amazon Machine Learning	`machinelearning`
AWS Marketplace	`aws-marketplace`
AWS Marketplace Management Portal	`aws-marketplace-management`
Amazon Mobile Analytics	`mobileanalytics`
AWS Mobile Hub	`mobilehub`
AWS OpsWorks	`opsworks`
AWS OpsWorks for Chef Automate	`opsworks-cm`
AWS Organizations	`organizations`
Amazon Polly	`polly`

Service	Namespace
Amazon Redshift	`redshift`
Amazon Relational Database Service (Amazon RDS)	`rds`
Amazon Route 53	`route53`
Amazon Route 53 Domains	`route53domains`
AWS Security Token Service (AWS STS)	`sts`
AWS Service Catalog	`servicecatalog`
Amazon Simple Email Service (Amazon SES)	`ses`
Amazon Simple Notification Service (Amazon SNS)	`sns`
Amazon Simple Queue Service (Amazon SQS)	`sqs`
Amazon Simple Storage Service (Amazon S3)	`s3`
Amazon Simple Workflow Service (Amazon SWF)	`swf`
Amazon SimpleDB	`sdb`
AWS Step Functions	`states`
AWS Storage Gateway	`storagegateway`
AWS Support	`support`
AWS Trusted Advisor	`trustedadvisor`
Amazon Virtual Private Cloud (Amazon VPC)	`ec2`
AWS WAF	`waf`
Amazon WorkMail	`workmail`
Amazon WorkSpaces	`workspaces`

Signing AWS API Requests

When you send HTTP requests to AWS, you sign the requests so that AWS can identify who sent them. You sign requests with your AWS access key, which consists of an access key ID and secret access key. Some requests do not need to be signed, such as anonymous requests to Amazon Simple Storage Service (Amazon S3) and some API operations in AWS Security Token Service (AWS STS) such as `AssumeRoleWithWebIdentity`.

Note
You need to learn how to sign HTTP requests only when you 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.

When Do You Need to Sign Requests?

When you write custom code to send HTTP requests to AWS, you need to include code to sign the requests. You might do this for the following reasons:

- You are working with a programming language for which there is no AWS SDK.
- You want complete control over how a request is sent to AWS.

You don't need to sign a request when you use the AWS Command Line Interface (AWS CLI) or one of the AWS SDKs. These tools manage the connection details, such as calculating signatures, handling request retries, and error handling. In most cases, they also contain sample code, tutorials, and other resources to help you get started writing applications that interact with AWS.

Why Requests Are Signed

The signing process helps secure requests in the following ways:

- **Verify the identity of the requester**

  Signing makes sure that the request has been sent by someone with a valid access key. For more information, see Understanding and Getting Your Security Credentials (p. 63).

- **Protect data in transit**

  To prevent tampering with a request while it's in transit, some of the request elements are used to calculate a hash (digest) of the request, and the resulting hash value is included as part of the request. When an AWS service receives the request, it uses the same information to calculate a hash and matches it against the hash value in your request. If the values don't match, AWS denies the request.
- **Protect against potential replay attacks**

  In most cases, a request must reach AWS within five minutes of the time stamp in the request. Otherwise, AWS denies the request.

Signing Requests

To sign a request, you calculate a hash (digest) of the request, and then use the hash value with some other values from the request and your access key to create a signed hash; this is the signature.

You add the signature to a request in one of the following ways:

- Add the signature to the request using the HTTP `Authorization` header.
- Add the signature as a query string value to the request. Because the request signature is part of the URL, this type of URL is called a presigned URL.

Signature Versions

AWS supports two signature versions: Signature Version 4 and Signature Version 2. You should use Signature Version 4. All AWS services support Signature Version 4, except Amazon SimpleDB which requires Signature Version 2. For AWS services that support both versions, we recommend that you use Signature Version 4.

All AWS regions support Signature Version 4.

Signature Version 4 Signing Process

Signature Version 4 is the process to add authentication information to AWS requests. For security, most requests to AWS must be signed with an access key, which consists of an access key ID and secret access key.

> **Important**
> 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. However, when you manually create HTTP requests to AWS, you must sign the requests yourself.

How Signature Version 4 works

1. You create a canonical request.
2. You use the canonical request and some other information to create a string to sign.
3. You use your AWS secret access key to derive a signing key, and then use that signing key and the string to sign to create a signature.
4. You add the resulting signature to the HTTP request in a header or as a query string parameter.

When AWS receives the request, it performs the same steps that you did to calculate the signature. AWS then compares the calculated signature to the one you sent with the request. If the signatures match, the request is processed. If the signatures don't match, the request is denied.

For more information, see the following resources:

- To get started with the signing process, see Signing AWS Requests with Signature Version 4 (p. 101).
- For sample signed requests, see Examples of the Complete Version 4 Signing Process (Python) (p. 114).
- If you have questions about Signature Version 4, post your question in the AWS Identity and Access Management forum.

Changes in Signature Version 4

Signature Version 4 is the current AWS signing protocol. It includes several changes from the previous Signature Version 2:

- To sign your message, you use a *signing key* that is derived from your secret access key rather than using the secret access key itself. For more information about deriving keys, see Task 3: Calculate the Signature for AWS Signature Version 4 (p. 107).
- You derive your signing key from the *credential scope*, which means that you don't need to include the key itself in the request. Credential scope is represented by a slash-separated string of dimensions in the following order:
  1. Date information as an eight-digit string representing the year (YYYY), month (MM), and day (DD) of the request (for example, `20150830`). For more information about handling dates, see Handling Dates in Signature Version 4 (p. 110).
  2. Region information as a lowercase alphanumeric string. Use the region name that is part of the service's endpoint. For services with a globally unique endpoint such as IAM, use `us-east-1`.
  3. Service name information as a lowercase alphanumeric string (for example, `iam`). Use the service name that is part of the service's endpoint. For example, the IAM endpoint is https://iam.amazonaws.com, so you use the string `iam` as part of the `Credential` parameter.
  4. A special termination string: `aws4_request`.

- You use the credential scope in each signing task:
  - If you add signing information to the query string, include the credential scope as part of the `X-Amz-Credential` parameter when you create the canonical request in Task 1: Create a Canonical Request for Signature Version 4 (p. 103).
  - You must include the credential scope as part of your string to sign in Task 2: Create a String to Sign for Signature Version 4 (p. 106).
  - Finally, you use the date, region, and service name components of the credential scope to derive your signing key in Task 3: Calculate the Signature for AWS Signature Version 4 (p. 107).

Signing AWS Requests with Signature Version 4

This section explains how to create a signature and add it to a request.

Topics

What Signing Looks Like in a Request

The following example shows what an HTTPS request might look like as it is sent from your client to AWS, without any signing information.

```
GET https://iam.amazonaws.com/?Action=ListUsers&Version=2010-05-08 HTTP/1.1
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Host: iam.amazonaws.com
X-Amz-Date: 20150830T123600Z
```

After you complete the signing tasks, you add the authentication information to the request. You can add the authentication information in two ways:

Authorization header

You can add the authentication information to the request with an `Authorization` header. Although the HTTP header is named `Authorization`, the signing information is actually used for authentication to establish who the request came from.

The `Authorization` header includes the following information:

- Algorithm you used for signing (AWS4-HMAC-SHA256)
- Credential scope (with your access key ID)
- List of signed headers
- Calculated signature. The signature is based on your request information, and you use your AWS secret access key to produce the signature. The signature confirms your identity to AWS.

The following example shows what the preceding request might look like after you've created the signing information and added it to the request in the `Authorization` header.

Note that in the actual request, the `Authorization` header would appear as a continuous line of text. The version below has been formatted for readability.

```
GET https://iam.amazonaws.com/?Action=ListUsers&Version=2010-05-08 HTTP/1.1
Authorization: AWS4-HMAC-SHA256
 Credential=AKIDEXAMPLE/20150830/us-east-1/iam/aws4_request,
 SignedHeaders=content-type;host;x-amz-date,
 Signature=5d672d79c15b13162d9279b0855cfba6789a8edb4c82c400e06b5924a6f2b5d7
content-type: application/x-www-form-urlencoded; charset=utf-8
host: iam.amazonaws.com
x-amz-date: 20150830T123600Z
```

Query string

As an alternative to adding authentication information with an HTTP request header, you can include it in the query string. The query string contains everything that is part of the request, including the name and parameters for the action, the date, and the authentication information.

The following example shows how you might construct a GET request with the action and authentication information in the query string.

(In the actual request, the query string would appear as a continuous line of text. The version below has been formatted with line breaks for readability.)

```
GET https://iam.amazonaws.com?Action=ListUsers&Version=2010-05-08
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIDEXAMPLE%2F20150830%2Fus-east-1%2Fiam%2Faws4_request
&X-Amz-Date=20150830T123600Z
&X-Amz-Expires=60
&X-Amz-SignedHeaders=content-type%3Bhost
&X-Amz-Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02 HTTP/1.1
content-type: application/x-www-form-urlencoded; charset=utf-8
host: iam.amazonaws.com
```

GET and POST Requests in the Query API

The query API that many AWS services support lets you make requests using either HTTP `GET` or `POST`. (In the query API, you can use `GET` even if you're making requests that change state; that is, the query API is not inherently RESTful.) Because `GET` requests pass parameters on the query string, they are limited to the maximum length of a URL. If a request includes a large payload (for example, you might upload a large IAM policy or send many parameters in JSON format for a DynamoDB request), you generally use a `POST` request.

The signing process is the same for both types of requests.

Summary of Signing Steps

To create a signed request, complete the following:

- Task 1: Create a Canonical Request for Signature Version 4 (p. 103)

  Arrange the contents of your request (host, action, headers, etc.) into a standard (*canonical*) format. The canonical request is one of the inputs used to create a string to sign.
- Task 2: Create a String to Sign for Signature Version 4 (p. 106)

  Create a *string to sign* with the canonical request and extra information such as the algorithm, request date, credential scope, and the digest (hash) of the canonical request.
- Task 3: Calculate the Signature for AWS Signature Version 4 (p. 107)

Derive a signing key by performing a succession of keyed hash operations (HMAC operations) on the request date, region, and service, with your AWS secret access key as the key for the initial hashing operation. After you derive the signing key, you then calculate the signature by performing a keyed hash operation on the string to sign. Use the derived signing key as the hash key for this operation.

- Task 4: Add the Signing Information to the Request (p. 109)

  After you calculate the signature, add it to an HTTP header or to the query string of the request.

> **Note**
> The AWS SDKs handle the signature calculation process for you, so you do not have to manually complete the signing process. For more information, see Tools for Amazon Web Services.

The following additional resources illustrate aspects of the signing process:

- Examples of How to Derive a Signing Key for Signature Version 4 (p. 111). This page shows how to derive a signing key using Java, C#, Python, Ruby, and JavaScript.
- Examples of the Complete Version 4 Signing Process (Python) (p. 114). This set of programs in Python provide complete examples of the signing process. The examples show signing with a `POST` request, with a `GET` request that has signing information in a request header, and with a `GET` request that has signing information in the query string.
- Signature Version 4 Test Suite (p. 121). This downloadable package contains a collection of examples that include signature information for various steps in the signing process. You can use these examples to verify that your signing code is producing the correct results at each step of the process.

Task 1: Create a Canonical Request for Signature Version 4

To begin the signing process, create a string that includes information from your request in a standardized (canonical) format. This ensures that when AWS receives the request, it can calculate the same signature that you calculated.

Follow the steps here to create a canonical version of the request. Otherwise, your version and the version calculated by AWS won't match, and the request will be denied.

The following example shows the pseudocode to create a canonical request.

In this pseudocode, `Hash` represents a function that produces a message digest, typically SHA-256. (Later in the process, you specify which hashing algorithm you're using.) `HexEncode` represents a function that returns the base-16 encoding of the digest in lowercase characters. For example, `HexEncode("m")` returns the value `6d` rather than `6D`. Each input byte must be represented as exactly two hexadecimal characters.

Signature Version 4 does not require that you use a particular character encoding to encode the canonical request. However, some AWS services might require a specific encoding. For more information, consult the documentation for that service.

The following examples show how to construct the canonical form of a request to IAM. The original request might look like this as it is sent from the client to AWS, except that this example does not include the signing information yet.

The preceding example request is a GET request (method) that makes a `ListUsers` API (action) call to AWS Identity and Access Management (host). This action takes the `Version` parameter.

To create a canonical request, concatenate the following components from each step into a single string:

1. Start with the HTTP request method (`GET`, `PUT`, `POST`, etc.), followed by a newline character.

2. Add the canonical URI parameter, followed by a newline character. The canonical URI is the URI-encoded version of the absolute path component of the URI, which is everything in the URI from the HTTP host to the question mark character ("?") that begins the query string parameters (if any).

   Normalize URI paths according to RFC 3986. Remove redundant and relative path components. Each path segment must be URI-encoded.

   > **Note**
   > In exception to this, you do not normalize URI paths for requests to Amazon S3. For example, if you have a bucket with an object named `my-object//example//photo.user`, use that path. Normalizing the path to `my-object/example/photo.user` will cause the request to fail. For more information, see Task 1: Create a Canonical Request in the *Amazon Simple Storage Service API Reference*.

   If the absolute path is empty, use a forward slash (/). In the example IAM request, nothing follows the host in the URI, so the absolute path is empty.

3. Add the canonical query string, followed by a newline character. If the request does not include a query string, use an empty string (essentially, a blank line). The example request has the following query string.

   To construct the canonical query string, complete the following steps:

   a. Sort the parameter names by character code point in ascending order. For example, a parameter name that begins with the uppercase letter F precedes a parameter name that begins with a lowercase letter b.
   b. URI-encode each parameter name and value according to the following rules:
      - Do not URI-encode any of the unreserved characters that RFC 3986 defines: A-Z, a-z, 0-9, hyphen (-), underscore (_), period (.), and tilde (~).
      - Percent-encode all other characters with %XY, where X and Y are hexadecimal characters (0-9 and uppercase A-F). For example, the space character must be encoded as %20 (not using '+', as some encoding schemes do) and extended UTF-8 characters must be in the form `%XY%ZA%BC`.
   c. Build the canonical query string by starting with the first parameter name in the sorted list.
   d. For each parameter, append the URI-encoded parameter name, followed by the equals sign character (=), followed by the URI-encoded parameter value. Use an empty string for parameters that have no value.
   e. Append the ampersand character (&) after each parameter value, except for the last value in the list.

   One option for the query API is to put all request parameters in the query string. For example, you can do this for Amazon S3 to create a presigned URL. In that case, the canonical query string must include not only parameters for the request, but also the parameters used as part of the signing process—the hashing algorithm, credential scope, date, and signed headers parameters.

   The following example shows a query string that includes authentication information. The example is formatted with line breaks for readability, but the canonical query string must be one continuous line of text in your code.

   For more information about authentication parameters, see Task 2: Create a String to Sign for Signature Version 4 (p. 106).

   > **Note**
   > You can use temporary security credentials provided by the AWS Security Token Service (AWS STS) to sign a request. The process is the same as using long-term credentials, but when you add signing information to the query string you must add an additional query parameter for the security token. The parameter name is `X-Amz-Security-Token`, and the

parameter's value is the URI-encoded session token (the string you received from AWS STS when you obtained temporary security credentials).
For some services, you must include the X-Amz-Security-Token query parameter in the canonical (signed) query string. For other services, you add the X-Amz-Security-Token parameter at the end, after you calculate the signature. For details, see the API reference documentation for that service.

4. Add the canonical headers, followed by a newline character. The canonical headers consist of a list of all the HTTP headers that you are including with the signed request.

   At a minimum, you must include the host header. Standard headers like content-type are optional. Different services might require other headers.

   To create the canonical headers list, convert all header names to lowercase and remove leading spaces and trailing spaces. Convert sequential spaces in the header value to a single space.

   The following pseudocode describes how to construct the canonical list of headers:

   ```
   CanonicalHeaders =
   CanonicalHeadersEntry0 + CanonicalHeadersEntry1 + ... + CanonicalHeadersEntryN
   CanonicalHeadersEntry =
   Lowercase(HeaderName) + ':' + Trimall(HeaderValue) + '\n'
   ```

   Lowercase represents a function that converts all characters to lowercase. The Trimall function removes excess white space before and after values, and converts sequential spaces to a single space.

   Build the canonical headers list by sorting the (lowercase) headers by character code and then iterating through the header names. Construct each header according to the following rules:

   - Append the lowercase header name followed by a colon.
   - Append a comma-separated list of values for that header. Do not sort the values in headers that have multiple values.
   - Append a new line ('\n').

   The following examples compare a more complex set of headers with their canonical form:

   > **Note**
   > Each header is followed by a newline character, meaning the complete list ends with a newline character.

   In the canonical form, the following changes were made:

   - The header names were converted to lowercase characters.
   - The headers were sorted by character code.
   - Leading and trailing spaces were removed from the my-header1 and my-header2 values.
   - Sequential spaces in a b c were converted to a single space for the my-header1 and my-header2 values.

   > **Note**
   > You can use temporary security credentials provided by the AWS Security Token Service (AWS STS) to sign a request. The process is the same as using long-term credentials, but when you include signing information in the Authorization header you must add an additional HTTP header for the security token. The header name is X-Amz-Security-Token, and the header's value is the session token (the string you received from AWS STS when you obtained temporary security credentials).

5. Add the signed headers, followed by a newline character. This value is the list of headers that you included in the canonical headers. By adding this list of headers, you tell AWS which headers in the request are part of the signing process and which ones AWS can ignore (for example, any additional headers added by a proxy) for purposes of validating the request.

   The `host` header must be included as a signed header. If you include a date or `x-amz-date` header, you must also include that header in the list of signed headers.

   To create the signed headers list, convert all header names to lowercase, sort them by character code, and use a semicolon to separate the header names. The following pseudocode describes how to construct a list of signed headers. `Lowercase` represents a function that converts all characters to lowercase.

   ```
   SignedHeaders =
   Lowercase(HeaderName0) + ';' + Lowercase(HeaderName1) + ";" + ... +
    Lowercase(HeaderNameN)
   ```

   Build the signed headers list by iterating through the collection of header names, sorted by lowercase character code. For each header name except the last, append a semicolon (';') to the header name to separate it from the following header name.
6. Use a hash (digest) function like SHA256 to create a hashed value from the payload in the body of the HTTP or HTTPS request. Signature Version 4 does not require that you use a particular character encoding to encode text in the payload. However, some AWS services might require a specific encoding. For more information, consult the documentation for that service.

   When you create the string to sign, you specify the signing algorithm that you used to hash the payload. For example, if you used SHA256, you will specify AWS4-HMAC-SHA256 as the signing algorithm. The hashed payload must be represented as a lowercase hexadecimal string.

   If the payload is empty, use an empty string as the input to the hash function. In the IAM example, the payload is empty.
7. To construct the finished canonical request, combine all the components from each step as a single string. As noted, each component ends with a newline character. If you follow the canonical request pseudocode explained earlier, the resulting canonical request is shown in the following example.
8. Create a digest (hash) of the canonical request with the same algorithm that you used to hash the payload.

   > **Note**
   > Signature Version 4 does not require that you use a particular character encoding to encode the canonical request before calculating the digest. However, some AWS services might require a specific encoding. For more information, consult the documentation for that service.

   The hashed canonical request must be represented as a string of lowercase hexademical characters. The following example shows the result of using SHA-256 to hash the example canonical request.

   You include the hashed canonical request as part of the string to sign in Task 2: Create a String to Sign for Signature Version 4 (p. 106).

Task 2: Create a String to Sign for Signature Version 4

The *string to sign* includes meta information about your request and about the canonical request that you created in Task 1: Create a Canonical Request for Signature Version 4 (p. 103). You will use the string to sign and a derived signing key that you create later as inputs to calculate the request signature in Task 3: Calculate the Signature for AWS Signature Version 4 (p. 107).

To create the string to sign, concatenate the algorithm, date and time, credential scope, and digest of the canonical request, as shown in the following pseudocode:

Structure of string to sign

```
StringToSign =
    Algorithm + \n +
    RequestDateTime + \n +
    CredentialScope + \n +
    HashedCanonicalRequest
```

The following example shows how to construct the string to sign with the same request from Task 1: Create A Canonical Request (p. 103).

To create the string to sign

1. Start with the algorithm designation, followed by a newline character. This value is the hashing algorithm that you use to calculate the digests in the canonical request. For SHA256, `AWS4-HMAC-SHA256` is the algorithm.

   ```
   AWS4-HMAC-SHA256\n
   ```

2. Append the request date value, followed by a newline character. The date is specified with ISO8601 basic format in the `x-amz-date` header in the format YYYYMMDD'T'HHMMSS'Z'. This value must match the value you used in any previous steps.

   ```
   20150830T123600Z\n
   ```

3. Append the credential scope value, followed by a newline character. This value is a string that includes the date, the region you are targeting, the service you are requesting, and a termination string ("`aws4_request`") in lowercase characters. The region and service name strings must be UTF-8 encoded.

   ```
   20150830/us-east-1/iam/aws4_request\n
   ```

   - The date must be in the `YYYYMMDD` format. Note that the date does not include a time value.
   - Verify that the region you specify is the region that you are sending the request to. See AWS Regions and Endpoints (p. 2).

4. Append the hash of the canonical request that you created in Task 1: Create a Canonical Request for Signature Version 4 (p. 103). This value is not followed by a newline character. The hashed canonical request must be lowercase base-16 encoded, as defined by Section 8 of RFC 4648.

   ```
   f536975d06c0309214f805bb90ccff089219ecd68b2577efef23edd43b7e1a59
   ```

The following string to sign is a request to IAM on August 30, 2015.

Task 3: Calculate the Signature for AWS Signature Version 4

Before you calculate a signature, you derive a signing key from your AWS secret access key. Because the derived signing key is specific to the date, service, and region, it offers a greater degree of protection. You don't just use your secret access key to sign the request. You then use the signing key and the string to sign that you created in Task 2: Create a String to Sign for Signature Version 4 (p. 106) as the inputs to a keyed hash function. The hex-encoded result from the keyed hash function is the signature.

Signature Version 4 does not require that you use a particular character encoding to encode the string to sign. However, some AWS services might require a specific encoding. For more information, consult the documentation for that service.

To calculate a signature

1. Derive your signing key. To do this, use your secret access key to create a series of hash-based message authentication codes (HMACs). This is shown in the following pseudocode, where `HMAC(key, data)` represents an HMAC-SHA256 function that returns output in binary format. The result of each hash function becomes input for the next one.

   Pseudocode for deriving a signing key

   ```
   kSecret = your secret access key
   kDate = HMAC("AWS4" + kSecret, Date)
   kRegion = HMAC(kDate, Region)
   kService = HMAC(kRegion, Service)
   kSigning = HMAC(kService, "aws4_request")
   ```

   Note that the date used in the hashing process is in the format `YYYYMMDD` (for example, `20150830`), and does not include the time.

   Make sure you specify the HMAC parameters in the correct order for the programming language you are using. This example shows the key as the first parameter and the data (message) as the second parameter, but the function that you use might specify the key and data in a different order.

   Use the digest (binary format) for the key derivation. Most languages have functions to compute either a binary format hash, commonly called a digest, or a hex-encoded hash, called a hexdigest. The key derivation requires that you use a binary-formatted digest.

   The following example show the inputs to derive a signing key and the resulting output, where kSecret = `wJalrXUtnFEMI/K7MDENG+bPxRfiCYEXAMPLEKEY`.

   The example uses the same parameters from the request in Task 1 and Task 2 (a request to IAM in the `us-east-1` region on August 30, 2015).

   Example inputs

   ```
   HMAC(HMAC(HMAC(HMAC("AWS4" + kSecret,"20150830"),"us-east-1"),"iam"),"aws4_request")
   ```

   The following example shows the derived signing key that results from this sequence of HMAC hash operations. This shows the hexadecimal representation of each byte in the binary signing key.

   Example signing key

   ```
   c4afb1cc5771d871763a393e44b703571b55cc28424d1a5e86da6ed3c154a4b9
   ```

   For more information about how to derive a signing key in different programming languages, see Examples of How to Derive a Signing Key for Signature Version 4 (p. 111).

2. Calculate the signature. To do this, use the signing key that you derived and the string to sign as inputs to the keyed hash function. After you calculate the signature as a digest, convert the binary value to a hexadecimal representation.

   The following pseudocode shows how to calculate the signature.

   ```
   signature = HexEncode(HMAC(derived signing key, string to sign))
   ```

   The following example shows the resulting signature if you use the same signing key and the string to sign from Task 2:

Example signature

```
5d672d79c15b13162d9279b0855cfba6789a8edb4c82c400e06b5924a6f2b5d7
```

Task 4: Add the Signing Information to the Request

After you calculate the signature, you add it to the request. You can add the signing information to a request in one of two ways:

- An HTTP header named `Authorization`
- The query string

You cannot pass signing information in both the `Authorization` header and the query string.

Note

You can use temporary security credentials provided by the AWS Security Token Service (AWS STS) to sign a request. The process is the same as using long-term credentials, but requires an additional HTTP header or query string parameter for the security token. The name of the header or query string parameter is `X-Amz-Security-Token`, and the value is the session token (the string you received from AWS STS when you obtained temporary security credentials).

When you add the `X-Amz-Security-Token` parameter to the query string, some services require that you include this parameter in the canonical (signed) request. For other services, you add this parameter at the end, after you calculate the signature. For details, see the API reference documentation for that service.

Adding Signing Information to the Authorization Header

You can include signing information by adding it to an HTTP header named `Authorization`. The contents of the header are created after you calculate the signature as described in the preceding steps, so the `Authorization` header is not included in the list of signed headers. Although the header is named `Authorization`, the signing information is actually used for authentication.

The following pseudocode shows the construction of the `Authorization` header.

```
Authorization: algorithm Credential=access key ID/credential scope,
 SignedHeaders=SignedHeaders, Signature=signature
```

The following example shows a finished `Authorization` header.

Note that in the actual request, the authorization header would appear as a continuous line of text. The version below has been formatted for readability.

```
Authorization: AWS4-HMAC-SHA256
Credential=AKIDEXAMPLE/20150830/us-east-1/iam/aws4_request,
SignedHeaders=content-type;host;x-amz-date,
Signature=5d672d79c15b13162d9279b0855cfba6789a8edb4c82c400e06b5924a6f2b5d7
```

Note the following:

- There is no comma between the algorithm and `Credential`. However, the `SignedHeaders` and `Signature` are separated from the preceding values with a comma.
- The `Credential` value starts with the access key ID, which is followed by a forward slash (`/`), which is followed by the credential scope that you calculated in Task 2: Create a String to Sign for Signature Version 4 (p. 106). The secret access key is used to derive the signing key for the signature, but is not included in the signing information sent in the request.

Adding Signing Information to the Query String

You can make requests and pass all request values in the query string, including signing information. This is sometimes referred to as a *presigned URL*, because it produces a single URL with everything required in order to make a successful call to AWS. It's commonly used in Amazon S3. For more information, see Authenticating Requests by Using Query Parameters (AWS Signature Version 4) in the *Amazon Simple Storage Service API Reference*.

> **Important**
> If you make a request in which all parameters are included in the query string, the resulting URL represents an AWS action that is already authenticated. Therefore, treat the resulting URL with as much caution as you would treat your actual credentials. We recommend you specify a short expiration time for the request with the `X-Amz-Expires` parameter.

When you use this approach, all the query string values (except the signature) are included in the canonical query string that is part of the canonical query that you construct in the first part of the signing process (p. 103).

The following pseudocode shows the construction of a query string that contains all request parameters.

```
querystring = Action=action
querystring += &X-Amz-Algorithm=algorithm
querystring += &X-Amz-Credential= urlencode(access_key_ID + '/' + credential_scope)
querystring += &X-Amz-Date=date
querystring += &X-Amz-Expires=timeout interval
querystring += &X-Amz-SignedHeaders=signed_headers
```

After the signature is calculated (which uses the other query string values as part of the calculation), you add the signature to the query string as the `X-Amz-Signature` parameter:

```
querystring += &X-Amz-Signature=signature
```

The following example shows what a request might look like when all the request parameters and the signing information are included in query string parameters.

Note that in the actual request, the authorization header would appear as a continuous line of text. The version below has been formatted for readability.

```
https://iam.amazonaws.com?Action=ListUsers&Version=2010-05-08
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIDEXAMPLE%2F20150830%2Fus-east-1%2Fiam%2Faws4_request
&X-Amz-Date=20150830T123600Z
&X-Amz-Expires=60
&X-Amz-SignedHeaders=content-type%3Bhost
&X-Amz-Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02
```

Note the following:

- For the signature calculation, query string parameters must be sorted in code point order from low to high, and their values must be URI-encoded. See the step about creating a canonical query string in Task 1: Create a Canonical Request for Signature Version 4 (p. 103).
- Set the timeout interval (`X-Amz-Expires`) to the minimal viable time for the operation you're requesting.

Handling Dates in Signature Version 4

The date that you use as part of your credential scope must match the date of your request. You can include the date as part of your request in several ways. You can use a `date` header, an `x-amz-date` header or include x-amz-date as a query parameter. For example requests, see Examples of the Complete Version 4 Signing Process (Python) (p. 114).

The time stamp must be in UTC and in the following ISO 8601 format: YYYYMMDD'T'HHMMSS'Z'. For example, 20150830T123600Z is a valid time stamp. Do not include milliseconds in the time stamp.

AWS first checks the x-amz-date header or parameter for a time stamp. If AWS can't find a value for x-amz-date, it looks for the date header. AWS then checks the credential scope for an eight-digit string representing the year (YYYY), month (MM), and day (DD) of the request. For example, if the x-amz-date header value is 20111015T080000Z and the date component of the credential scope is 20111015, AWS allows the authentication process to proceed.

If the dates don't match, AWS rejects the request, even if the time stamp is only seconds away from the date in the credential scope. For example, AWS will reject a request that has an x-amz-date header value of 20151014T235959Z and a credential scope that has the date 20151015.

Examples of How to Derive a Signing Key for Signature Version 4

This page shows examples in several programming languages for how to derive a signing key for Signature Version 4. The examples on this page show only how to derive a signing key, which is just one part of signing AWS requests. For examples that show the complete process, see Examples of the Complete Version 4 Signing Process (Python) (p. 114).

> **Note**
> If you are using one of the AWS SDKs (including the SDK for Java, .NET, Python, Ruby, or JavaScript), you do not have to manually perform the steps of deriving a signing key and adding authentication information to a request. The SDKs perform this work for you. You need to manually sign requests only if you are directly making HTTP or HTTPS requests.

Topics

- Deriving the Signing Key with Java (p. 111)
- Deriving the Signing Key with .NET (C#) (p. 112)
- Deriving the Signing Key with Python (p. 112)
- Deriving the Signing Key with Ruby (p. 112)
- Deriving the Signing Key with JavaScript (p. 112)
- Deriving the Signing Key with Other Languages (p. 113)
- Common Coding Mistakes (p. 113)

Deriving the Signing Key with Java

```
static byte[] HmacSHA256(String data, byte[] key) throws Exception {
    String algorithm="HmacSHA256";
    Mac mac = Mac.getInstance(algorithm);
    mac.init(new SecretKeySpec(key, algorithm));
    return mac.doFinal(data.getBytes("UTF8"));
}

static byte[] getSignatureKey(String key, String dateStamp, String regionName, String
 serviceName) throws Exception {
    byte[] kSecret = ("AWS4" + key).getBytes("UTF8");
    byte[] kDate = HmacSHA256(dateStamp, kSecret);
    byte[] kRegion = HmacSHA256(regionName, kDate);
    byte[] kService = HmacSHA256(serviceName, kRegion);
    byte[] kSigning = HmacSHA256("aws4_request", kService);
    return kSigning;
}
```

Deriving the Signing Key with .NET (C#)

```
static byte[] HmacSHA256(String data, byte[] key)
{
    String algorithm = "HmacSHA256";
    KeyedHashAlgorithm kha = KeyedHashAlgorithm.Create(algorithm);
    kha.Key = key;

    return kha.ComputeHash(Encoding.UTF8.GetBytes(data));
}

static byte[] getSignatureKey(String key, String dateStamp, String regionName, String
 serviceName)
{
    byte[] kSecret = Encoding.UTF8.GetBytes(("AWS4" + key).ToCharArray());
    byte[] kDate = HmacSHA256(dateStamp, kSecret);
    byte[] kRegion = HmacSHA256(regionName, kDate);
    byte[] kService = HmacSHA256(serviceName, kRegion);
    byte[] kSigning = HmacSHA256("aws4_request", kService);

    return kSigning;
}
```

Deriving the Signing Key with Python

```
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()

def getSignatureKey(key, dateStamp, regionName, serviceName):
    kDate = sign(("AWS4" + key).encode("utf-8"), dateStamp)
    kRegion = sign(kDate, regionName)
    kService = sign(kRegion, serviceName)
    kSigning = sign(kService, "aws4_request")
    return kSigning
```

Deriving the Signing Key with Ruby

```
def getSignatureKey key, dateStamp, regionName, serviceName
    kDate = OpenSSL::HMAC.digest('sha256', "AWS4" + key, dateStamp)
    kRegion = OpenSSL::HMAC.digest('sha256', kDate, regionName)
    kService = OpenSSL::HMAC.digest('sha256', kRegion, serviceName)
    kSigning = OpenSSL::HMAC.digest('sha256', kService, "aws4_request")

    kSigning
end
```

Deriving the Signing Key with JavaScript

The following example uses the crypto-js library. For more information, see https://www.npmjs.com/package/crypto-js and https://code.google.com/archive/p/crypto-js/.

```
var crypto = require("crypto-js");
```

```
function getSignatureKey(Crypto, key, dateStamp, regionName, serviceName) {
    var kDate = Crypto.HmacSHA256(dateStamp, "AWS4" + key);
    var kRegion = Crypto.HmacSHA256(regionName, kDate);
    var kService = Crypto.HmacSHA256(serviceName, kRegion);
    var kSigning = Crypto.HmacSHA256("aws4_request", kService);
    return kSigning;
}
```

Deriving the Signing Key with Other Languages

If you need to implement this logic in a different programming language, we recommend testing the intermediary steps of the key derivation algorithm against the values in this section. The following example in Ruby prints the results using the `hexEncode` function after each step in the algorithm.

```
def hexEncode bindata
    result=""
    data=bindata.unpack("C*")
    data.each {|b| result+= "%02x" % b}
    result
end
```

Given the following test input:

```
key = 'wJalrXUtnFEMI/K7MDENG+bPxRfiCYEXAMPLEKEY'
dateStamp = '20120215'
regionName = 'us-east-1'
serviceName = 'iam'
```

Your program should generate the following values for the values in `getSignatureKey`. Note that these are hex-encoded representations of the binary data; the key itself and the intermediate values should be in binary format.

```
kSecret  =
 '41575334774a616c725855746e46454d492f4b374d44454e472b62507852666943594558414d504c454b4559'
kDate    = '969fbb94feb542b71ede6f87fe4d5fa29c789342b0f407474670f0c2489e0a0d'
kRegion  = '69daa0209cd9c5ff5c8ced464a696fd4252e981430b10e3d3fd8e2f197d7a70c'
kService = 'f72cfd46f26bc4643f06a11eabb6c0ba18780c19a8da0c31ace671265e3c87fa'
kSigning = 'f4780e2d9f65fa895f9c67b32ce1baf0b0d8a43505a000a1a9e090d414db404d'
```

Common Coding Mistakes

To simplify your task, avoid the following common coding errors.

Tip
Examine the HTTP request that you're sending to AWS with a tool that shows you what your raw HTTP requests look like. This can help you spot issues that aren't evident from your code.

- Don't include an extra newline character, or forget one where it's required.
- Don't format the date incorrectly in the credential scope, such as using a time stamp instead of YYYYMMDD format.
- Make sure the headers in the canonical headers and the signed headers are the same.
- Don't inadvertently swap the key and the data (message) when calculating intermediary keys. The result of the previous step's computation is the key, not the data. Check the documentation for your cryptographic primitives carefully to ensure that you place the parameters in the proper order.

- Don't forget to add the string "AWS4" in front of the key for the first step. If you implement the key derivation using a `for` loop or iterator, don't forget to special-case the first iteration so that it includes the "AWS4" string.

For more information about possible errors, see Troubleshooting AWS Signature Version 4 Errors (p. 123).

Examples of the Complete Version 4 Signing Process (Python)

This section shows example programs written in Python that illustrate how to work with Signature Version 4 in AWS. We deliberately wrote these example programs to be simple (to use few Python-specific features) to make it easier to understand the overall process of signing AWS requests.

In order to work with these example programs, you need the following:

- Python 2.x installed on your computer, which you can get from the Python site. These programs were tested using Python 2.7.
- The Python **requests** library, which is used in the example script to make web requests. A convenient way to install Python packages is to use `pip`, which gets packages from the Python package index site. You can then install `requests` by running `pip install requests` at the command line.
- An access key (access key ID and secret access key) in environment variables named `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`. Alternatively, you can keep these values in a credentials file and read them from that file. As a best practice, we recommend that you do ***not*** embed credentials in code. For more information, see Best Practices for Managing AWS Access Keys in the *Amazon Web Services General Reference*.

> **Note**
> The following examples use UTF-8 to encode the canonical request and string to sign, but Signature Version 4 does not require that you use a particular character encoding. However, some AWS services might require a specific encoding. For more information, consult the documentation for that service.

Topics

Using GET with an Authorization Header (Python)

The following example shows how to make a request using the Amazon EC2 query API. The request makes a GET request and passes authentication information to AWS using the `Authorization` header.

```
# AWS Version 4 signing example

# EC2 API (DescribeRegions)

# See: http://docs.aws.amazon.com/general/latest/gr/sigv4_signing.html
# This version makes a GET request and passes the signature
# in the Authorization header.
import sys, os, base64, datetime, hashlib, hmac
import requests # pip install requests
```

```
# ************* REQUEST VALUES *************
method = 'GET'
service = 'ec2'
host = 'ec2.amazonaws.com'
region = 'us-east-1'
endpoint = 'https://ec2.amazonaws.com'
request_parameters = 'Action=DescribeRegions&Version=2013-10-15'

# Key derivation functions. See:
# http://docs.aws.amazon.com/general/latest/gr/signature-v4-examples.html#signature-v4-
examples-python
def sign(key, msg):
    return hmac.new(key, msg.encode('utf-8'), hashlib.sha256).digest()

def getSignatureKey(key, dateStamp, regionName, serviceName):
    kDate = sign(('AWS4' + key).encode('utf-8'), dateStamp)
    kRegion = sign(kDate, regionName)
    kService = sign(kRegion, serviceName)
    kSigning = sign(kService, 'aws4_request')
    return kSigning

# Read AWS access key from env. variables or configuration file. Best practice is NOT
# to embed credentials in code.
access_key = os.environ.get('AWS_ACCESS_KEY_ID')
secret_key = os.environ.get('AWS_SECRET_ACCESS_KEY')
if access_key is None or secret_key is None:
    print 'No access key is available.'
    sys.exit()

# Create a date for headers and the credential string
t = datetime.datetime.utcnow()
amzdate = t.strftime('%Y%m%dT%H%M%SZ')
datestamp = t.strftime('%Y%m%d') # Date w/o time, used in credential scope


# ************* TASK 1: CREATE A CANONICAL REQUEST *************
# http://docs.aws.amazon.com/general/latest/gr/sigv4-create-canonical-request.html

# Step 1 is to define the verb (GET, POST, etc.)--already done.

# Step 2: Create canonical URI--the part of the URI from domain to query
# string (use '/' if no path)
canonical_uri = '/'

# Step 3: Create the canonical query string. In this example (a GET request),
# request parameters are in the query string. Query string values must
# be URL-encoded (space=%20). The parameters must be sorted by name.
# For this example, the query string is pre-formatted in the request_parameters variable.
canonical_querystring = request_parameters

# Step 4: Create the canonical headers and signed headers. Header names
# must be trimmed and lowercase, and sorted in code point order from
# low to high. Note that there is a trailing \n.
canonical_headers = 'host:' + host + '\n' + 'x-amz-date:' + amzdate + '\n'

# Step 5: Create the list of signed headers. This lists the headers
# in the canonical_headers list, delimited with ";" and in alpha order.
# Note: The request can include any headers; canonical_headers and
# signed_headers lists those that you want to be included in the
# hash of the request. "Host" and "x-amz-date" are always required.
signed_headers = 'host;x-amz-date'

# Step 6: Create payload hash (hash of the request body content). For GET
# requests, the payload is an empty string ("").
payload_hash = hashlib.sha256('').hexdigest()
```

```
# Step 7: Combine elements to create create canonical request
canonical_request = method + '\n' + canonical_uri + '\n' + canonical_querystring + '\n' +
 canonical_headers + '\n' + signed_headers + '\n' + payload_hash


# ************* TASK 2: CREATE THE STRING TO SIGN*************
# Match the algorithm to the hashing algorithm you use, either SHA-1 or
# SHA-256 (recommended)
algorithm = 'AWS4-HMAC-SHA256'
credential_scope = datestamp + '/' + region + '/' + service + '/' + 'aws4_request'
string_to_sign = algorithm + '\n' +  amzdate + '\n' +  credential_scope + '\n' +
 hashlib.sha256(canonical_request).hexdigest()


# ************* TASK 3: CALCULATE THE SIGNATURE *************
# Create the signing key using the function defined above.
signing_key = getSignatureKey(secret_key, datestamp, region, service)

# Sign the string_to_sign using the signing_key
signature = hmac.new(signing_key, (string_to_sign).encode('utf-8'),
 hashlib.sha256).hexdigest()


# ************* TASK 4: ADD SIGNING INFORMATION TO THE REQUEST *************
# The signing information can be either in a query string value or in
# a header named Authorization. This code shows how to use a header.
# Create authorization header and add to request headers
authorization_header = algorithm + ' ' + 'Credential=' + access_key + '/' +
 credential_scope + ', ' +  'SignedHeaders=' + signed_headers + ', ' + 'Signature=' +
 signature

# The request can include any headers, but MUST include "host", "x-amz-date",
# and (for this scenario) "Authorization". "host" and "x-amz-date" must
# be included in the canonical_headers and signed_headers, as noted
# earlier. Order here is not significant.
# Python note: The 'host' header is added automatically by the Python 'requests' library.
headers = {'x-amz-date':amzdate, 'Authorization':authorization_header}


# ************* SEND THE REQUEST *************
request_url = endpoint + '?' + canonical_querystring

print '\nBEGIN REQUEST++++++++++++++++++++++++++++++++++++'
print 'Request URL = ' + request_url
r = requests.get(request_url, headers=headers)

print '\nRESPONSE++++++++++++++++++++++++++++++++++++'
print 'Response code: %d\n' % r.status_code
print r.text
```

Using POST (Python)

The following example shows how to make a request using the Amazon DynamoDB query API. The request makes a POST request and passes values to AWS in the body of the request. Authentication information is passed using the `Authorization` request header.

```
# AWS Version 4 signing example

# DynamoDB API (CreateTable)

# See: http://docs.aws.amazon.com/general/latest/gr/sigv4_signing.html
# This version makes a POST request and passes request parameters
# in the body (payload) of the request. Auth information is passed in
# an Authorization header.
```

```
import sys, os, base64, datetime, hashlib, hmac
import requests # pip install requests

# ************* REQUEST VALUES *************
method = 'POST'
service = 'dynamodb'
host = 'dynamodb.us-west-2.amazonaws.com'
region = 'us-west-2'
endpoint = 'https://dynamodb.us-west-2.amazonaws.com/'
# POST requests use a content type header. For DynamoDB,
# the content is JSON.
content_type = 'application/x-amz-json-1.0'
# DynamoDB requires an x-amz-target header that has this format:
#     DynamoDB_<API version>.<operationName>
amz_target = 'DynamoDB_20120810.CreateTable'

# Request parameters for CreateTable--passed in a JSON block.
request_parameters =  '{'
request_parameters +=  '"KeySchema": [{"KeyType": "HASH","AttributeName": "Id"}],'
request_parameters +=  '"TableName": "TestTable","AttributeDefinitions": [{"AttributeName":
 "Id","AttributeType": "S"}],'
request_parameters +=  '"ProvisionedThroughput": {"WriteCapacityUnits":
 5,"ReadCapacityUnits": 5}'
request_parameters +=  '}'

# Key derivation functions. See:
# http://docs.aws.amazon.com/general/latest/gr/signature-v4-examples.html#signature-v4-
examples-python
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()

def getSignatureKey(key, date_stamp, regionName, serviceName):
    kDate = sign(('AWS4' + key).encode('utf-8'), date_stamp)
    kRegion = sign(kDate, regionName)
    kService = sign(kRegion, serviceName)
    kSigning = sign(kService, 'aws4_request')
    return kSigning

# Read AWS access key from env. variables or configuration file. Best practice is NOT
# to embed credentials in code.
access_key = os.environ.get('AWS_ACCESS_KEY_ID')
secret_key = os.environ.get('AWS_SECRET_ACCESS_KEY')
if access_key is None or secret_key is None:
    print 'No access key is available.'
    sys.exit()

# Create a date for headers and the credential string
t = datetime.datetime.utcnow()
amz_date = t.strftime('%Y%m%dT%H%M%SZ')
date_stamp = t.strftime('%Y%m%d') # Date w/o time, used in credential scope


# ************* TASK 1: CREATE A CANONICAL REQUEST *************
# http://docs.aws.amazon.com/general/latest/gr/sigv4-create-canonical-request.html

# Step 1 is to define the verb (GET, POST, etc.)--already done.

# Step 2: Create canonical URI--the part of the URI from domain to query
# string (use '/' if no path)
canonical_uri = '/'

## Step 3: Create the canonical query string. In this example, request
# parameters are passed in the body of the request and the query string
# is blank.
canonical_querystring = ''
```

```
# Step 4: Create the canonical headers. Header names must be trimmed
# and lowercase, and sorted in code point order from low to high.
# Note that there is a trailing \n.
canonical_headers = 'content-type:' + content_type + '\n' + 'host:' + host + '\n' + 'x-amz-
date:' + amz_date + '\n' + 'x-amz-target:' + amz_target + '\n'

# Step 5: Create the list of signed headers. This lists the headers
# in the canonical_headers list, delimited with ";" and in alpha order.
# Note: The request can include any headers; canonical_headers and
# signed_headers include those that you want to be included in the
# hash of the request. "Host" and "x-amz-date" are always required.
# For DynamoDB, content-type and x-amz-target are also required.
signed_headers = 'content-type;host;x-amz-date;x-amz-target'

# Step 6: Create payload hash. In this example, the payload (body of
# the request) contains the request parameters.
payload_hash = hashlib.sha256(request_parameters).hexdigest()

# Step 7: Combine elements to create create canonical request
canonical_request = method + '\n' + canonical_uri + '\n' + canonical_querystring + '\n' +
 canonical_headers + '\n' + signed_headers + '\n' + payload_hash


# ************* TASK 2: CREATE THE STRING TO SIGN*************
# Match the algorithm to the hashing algorithm you use, either SHA-1 or
# SHA-256 (recommended)
algorithm = 'AWS4-HMAC-SHA256'
credential_scope = date_stamp + '/' + region + '/' + service + '/' + 'aws4_request'
string_to_sign = algorithm + '\n' +  amz_date + '\n' +  credential_scope + '\n' +
 hashlib.sha256(canonical_request).hexdigest()


# ************* TASK 3: CALCULATE THE SIGNATURE *************
# Create the signing key using the function defined above.
signing_key = getSignatureKey(secret_key, date_stamp, region, service)

# Sign the string_to_sign using the signing_key
signature = hmac.new(signing_key, (string_to_sign).encode('utf-8'),
 hashlib.sha256).hexdigest()


# ************* TASK 4: ADD SIGNING INFORMATION TO THE REQUEST *************
# Put the signature information in a header named Authorization.
authorization_header = algorithm + ' ' + 'Credential=' + access_key + '/' +
 credential_scope + ', ' +  'SignedHeaders=' + signed_headers + ', ' + 'Signature=' +
 signature

# For DynamoDB, the request can include any headers, but MUST include "host", "x-amz-date",
# "x-amz-target", "content-type", and "Authorization". Except for the authorization
# header, the headers must be included in the canonical_headers and signed_headers values,
 as
# noted earlier. Order here is not significant.
# # Python note: The 'host' header is added automatically by the Python 'requests' library.
headers = {'Content-Type':content_type,
           'X-Amz-Date':amz_date,
           'X-Amz-Target':amz_target,
           'Authorization':authorization_header}


# ************* SEND THE REQUEST *************
print '\nBEGIN REQUEST++++++++++++++++++++++++++++++++++++'
print 'Request URL = ' + endpoint

r = requests.post(endpoint, data=request_parameters, headers=headers)

print '\nRESPONSE++++++++++++++++++++++++++++++++++++'
print 'Response code: %d\n' % r.status_code
print r.text
```

Using GET with Authentication Information in the Query String (Python)

The following example shows how to make a request using the IAM query API. The request makes a GET request and passes parameters and signing information using the query string.

```
# AWS Version 4 signing example

# IAM API (CreateUser)

# See: http://docs.aws.amazon.com/general/latest/gr/sigv4_signing.html
# This version makes a GET request and passes request parameters
# and authorization information in the query string
import sys, os, base64, datetime, hashlib, hmac, urllib
import requests # pip install requests

# ************* REQUEST VALUES *************
method = 'GET'
service = 'iam'
host = 'iam.amazonaws.com'
region = 'us-east-1'
endpoint = 'https://iam.amazonaws.com'

# Key derivation functions. See:
# http://docs.aws.amazon.com/general/latest/gr/signature-v4-examples.html#signature-v4-
examples-python
def sign(key, msg):
    return hmac.new(key, msg.encode('utf-8'), hashlib.sha256).digest()

def getSignatureKey(key, dateStamp, regionName, serviceName):
    kDate = sign(('AWS4' + key).encode('utf-8'), dateStamp)
    kRegion = sign(kDate, regionName)
    kService = sign(kRegion, serviceName)
    kSigning = sign(kService, 'aws4_request')
    return kSigning

# Read AWS access key from env. variables or configuration file. Best practice is NOT
# to embed credentials in code.
access_key = os.environ.get('AWS_ACCESS_KEY_ID')
secret_key = os.environ.get('AWS_SECRET_ACCESS_KEY')
if access_key is None or secret_key is None:
    print 'No access key is available.'
    sys.exit()

# Create a date for headers and the credential string
t = datetime.datetime.utcnow()
amz_date = t.strftime('%Y%m%dT%H%M%SZ') # Format date as YYYYMMDD'T'HHMMSS'Z'
datestamp = t.strftime('%Y%m%d') # Date w/o time, used in credential scope


# ************* TASK 1: CREATE A CANONICAL REQUEST *************
# http://docs.aws.amazon.com/general/latest/gr/sigv4-create-canonical-request.html

# Because almost all information is being passed in the query string,
# the order of these steps is slightly different than examples that
# use an authorization header.

# Step 1: Define the verb (GET, POST, etc.)--already done.

# Step 2: Create canonical URI--the part of the URI from domain to query
# string (use '/' if no path)
canonical_uri = '/'

# Step 3: Create the canonical headers and signed headers. Header names
# must be trimmed and lowercase, and sorted in code point order from
# low to high. Note trailing \n in canonical_headers.
# signed_headers is the list of headers that are being included
# as part of the signing process. For requests that use query strings,
# only "host" is included in the signed headers.
canonical_headers = 'host:' + host + '\n'
signed_headers = 'host'

# Match the algorithm to the hashing algorithm you use, either SHA-1 or
# SHA-256 (recommended)
algorithm = 'AWS4-HMAC-SHA256'
credential_scope = datestamp + '/' + region + '/' + service + '/' + 'aws4_request'

# Step 4: Create the canonical query string. In this example, request
# parameters are in the query string. Query string values must
# be URL-encoded (space=%20). The parameters must be sorted by name.
canonical_querystring = 'Action=CreateUser&UserName=NewUser&Version=2010-05-08'
canonical_querystring += '&X-Amz-Algorithm=AWS4-HMAC-SHA256'
canonical_querystring += '&X-Amz-Credential=' + urllib.quote_plus(access_key + '/' +
 credential_scope)
canonical_querystring += '&X-Amz-Date=' + amz_date
canonical_querystring += '&X-Amz-Expires=30'
canonical_querystring += '&X-Amz-SignedHeaders=' + signed_headers

# Step 5: Create payload hash. For GET requests, the payload is an
# empty string ("").
payload_hash = hashlib.sha256('').hexdigest()

# Step 6: Combine elements to create create canonical request
canonical_request = method + '\n' + canonical_uri + '\n' + canonical_querystring + '\n' +
 canonical_headers + '\n' + signed_headers + '\n' + payload_hash


# ************* TASK 2: CREATE THE STRING TO SIGN*************
string_to_sign = algorithm + '\n' +  amz_date + '\n' +  credential_scope + '\n' +
 hashlib.sha256(canonical_request).hexdigest()


# ************* TASK 3: CALCULATE THE SIGNATURE *************
# Create the signing key
signing_key = getSignatureKey(secret_key, datestamp, region, service)

# Sign the string_to_sign using the signing_key
signature = hmac.new(signing_key, (string_to_sign).encode("utf-8"),
 hashlib.sha256).hexdigest()


# ************* TASK 4: ADD SIGNING INFORMATION TO THE REQUEST *************
# The auth information can be either in a query string
# value or in a header named Authorization. This code shows how to put
# everything into a query string.
canonical_querystring += '&X-Amz-Signature=' + signature


# ************* SEND THE REQUEST *************
# The 'host' header is added automatically by the Python 'request' lib. But it
# must exist as a header in the request.
request_url = endpoint + "?" + canonical_querystring

print '\nBEGIN REQUEST++++++++++++++++++++++++++++++++++++'
print 'Request URL = ' + request_url
r = requests.get(request_url)
```

```
print '\nRESPONSE++++++++++++++++++++++++++++++++++++'
print 'Response code: %d\n' % r.status_code
print r.text
```

Signature Version 4 Test Suite

To assist you in the development of an AWS client that supports Signature Version 4, you can use the files in the test suite to ensure your code is performing each step of the signing process correctly.

You can download the test suite at https://s3.amazonaws.com/awsdocs/aws-sig-v4-test-suite.zip.

Topics

Each test group contains five files that you can use to validate each of the tasks described in Signature Version 4 Signing Process (p. 100). The following list describes the contents of each file.

- `<file-name>.req`—the web request to be signed.
- `<file-name>.creq`—the resulting canonical request.
- `<file-name>.sts`—the resulting string to sign.
- `<file-name>.authz`—the `Authorization` header.
- `<file-name>.sreq`— the signed request.

Credential Scope and Secret Key

The examples in the test suite use the following credential scope:

```
AKIDEXAMPLE/20150830/us-east-1/service/aws4_request
```

The example secret key used for signing is:

```
wJalrXUtnFEMI/K7MDENG+bPxRfiCYEXAMPLEKEY
```

Example—A Simple GET Request with Parameters

The following example shows the web request to be signed from the `get-vanilla-query-order-key-case.req` file. This is the original request.

```
GET /?Param2=value2&Param1=value1 HTTP/1.1
Host:example.amazonaws.com
X-Amz-Date:20150830T123600Z
```

Task 1: Create a Canonical Request

In the steps outlined in Task 1: Create a Canonical Request for Signature Version 4 (p. 103), change the request in the `get-vanilla-query-order-key-case.req` file.

```
GET /?Param2=value2&Param1=value1 HTTP/1.1
Host:example.amazonaws.com
```

```
X-Amz-Date:20150830T123600Z
```

This creates the canonical request in the `get-vanilla-query-order-key-case.creq` file.

```
GET
/
Param1=value1&Param2=value2
host:example.amazonaws.com
x-amz-date:20150830T123600Z

host;x-amz-date
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Notes

- The parameters are sorted alphabetically (by character code).
- The header names are lowercase.
- There is a line break between the `x-amz-date` header and the signed headers.
- The hash of the payload is the hash of the empty string.

Task 2: Create a String to Sign

The hash of the canonical request returns the following value:

```
816cd5b414d056048ba4f7c5386d6e0533120fb1fcfa93762cf0fc39e2cf19e0
```

In the steps outlined in Task 2: Create a String to Sign for Signature Version 4 (p. 106), add the algorithm, request date, credential scope, and the canonical request hash to create the string to sign.

The result is the `get-vanilla-query-order-key-case.sts` file.

```
AWS4-HMAC-SHA256
20150830T123600Z
20150830/us-east-1/service/aws4_request
816cd5b414d056048ba4f7c5386d6e0533120fb1fcfa93762cf0fc39e2cf19e0
```

Notes

- The date on the second line matches the `x-amz-date` header, as well as the first element in the credential scope.
- The last line is the hex-encoded value for the hash of the canonical request.

Task 3: Calculate the Signature

In the steps outlined in Task 3: Calculate the Signature for AWS Signature Version 4 (p. 107), create a signature with your signing key and the string to sign from the `get-vanilla-query-order-key-case.sts file`.

The result generates the contents in the `get-vanilla-query-order-key-case.authz` file.

```
AWS4-HMAC-SHA256 Credential=AKIDEXAMPLE/20150830/us-east-1/
service/aws4_request, SignedHeaders=host;x-amz-date,
 Signature=b97d918cfa904a5beff61c982a1b6f458b799221646efd99d3219ec94cdf2500
```

Task 4: Add the Signing Information to the Request

In the steps outlined in Task 4: Add the Signing Information to the Request (p. 109), add the signing information generated in task 3 to the original request. For example, take the contents in the `get-vanilla-query-order-key-case.authz`, add it to the `Authorization` header, and then add the result to the `get-vanilla-query-order-key-case.req`.

This creates the signed request in the `get-vanilla-query-order-key-case.sreq` file.

```
GET /?Param2=value2&Param1=value1 HTTP/1.1
Host:example.amazonaws.com
X-Amz-Date:20150830T123600Z
Authorization: AWS4-HMAC-SHA256 Credential=AKIDEXAMPLE/20150830/
us-east-1/service/aws4_request, SignedHeaders=host;x-amz-date,
 Signature=b97d918cfa904a5beff61c982a1b6f458b799221646efd99d3219ec94cdf2500
```

Troubleshooting AWS Signature Version 4 Errors

Topics

- Troubleshooting AWS Signature Version 4 Canonicalization Errors (p. 123)
- Troubleshooting AWS Signature Version 4 Credential Scope Errors (p. 124)
- Troubleshooting AWS Signature Version 4 Key Signing Errors (p. 125)

When you develop code that implements Signature Version 4, you might receive errors from AWS products that you test against. The errors typically come from an error in the canonicalization of the request, the incorrect derivation or use of the signing key, or a validation failure of signature-specific parameters sent along with the request.

Troubleshooting AWS Signature Version 4 Canonicalization Errors

Consider the following request:

```
https://iam.amazonaws.com/?MaxItems=100
&Action=ListGroupsForUser
&UserName=Test
&Version=2010-05-08
&X-Amz-Date=20120223T063000Z
&X-Amz-Algorithm=AWS4-HMAC-SHA256
&X-Amz-Credential=AKIAIOSFODNN7EXAMPLE/20120223/us-east-1/iam/aws4_request
&X-Amz-SignedHeaders=host
&X-Amz-Signature=<calculated value>
```

If you incorrectly calculate the canonical request or the string to sign, the signature verification step performed by the service fails. The following example is a typical error response, which includes the canonical string and the string to sign as computed by the service. You can troubleshoot your calculation error by comparing the returned strings with the canonical string and your calculated string to sign.

```
<ErrorResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>SignatureDoesNotMatch</Code>
    <Message>The request signature we calculated does not match the signature you provided.
 Check your AWS Secret Access Key and signing method. Consult the service documentation for
 details.
```

```
The canonical string for this request should have been 'GET /
 Action=ListGroupsForUser&MaxItems=100&UserName=Test&Version=2010-05-08&X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential
=AKIAIOSFODNN7EXAMPLE%2F20120223%2Fus-east-1%2Fiam%2Faws4_request&X-Amz-
Date=20120223T063000Z&X-Amz-SignedHeaders=host
host:iam.amazonaws.com

host
<hashed-value>'

The String-to-Sign should have been
'AWS4-HMAC-SHA256
20120223T063000Z
20120223/us-east-1/iam/aws4_request
<hashed-value>'
</Message>
  </Error>
  <RequestId>4ced6e96-5de8-11e1-aa78-a56908bdf8eb</RequestId>
</ErrorResponse>
```

For testing with an SDK, we recommend troubleshooting by verifying each derivation step against known values. For more information, see Signature Version 4 Test Suite (p. 121).

Troubleshooting AWS Signature Version 4 Credential Scope Errors

AWS products validate credentials for proper scope; the credential parameter must specify the correct service, region, and date. For example, the following credential references the Amazon RDS service:

```
Credential=AKIAIOSFODNN7EXAMPLE/20120224/us-east-1/rds/aws4_request
```

If you use the same credentials to submit a request to IAM, you'll receive the following error response:

```
<ErrorResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>SignatureDoesNotMatch</Code>
    <Message>Credential should be scoped to correct service: 'iam'. </Message>
  </Error>
  <RequestId>aa0da9de-5f2b-11e1-a2c0-c1dc98b6c575</RequestId>
```

The credential must also specify the correct region. For example, the following credential for an IAM request incorrectly specifies the US West (N. California) region.

```
Credential=AKIAIOSFODNN7EXAMPLE/20120224/us-west-1/iam/aws4_request
```

If you use the credential to submit a request to IAM, which accepts only the `us-east-1` region specification, you'll receive the following response:

```
<ErrorResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>SignatureDoesNotMatch</Code>
    <Message>Credential should be scoped to a valid region, not 'us-east-1'. </Message>
  </Error>
  <RequestId>8e229682-5f27-11e1-88f2-4b1b00f424ae</RequestId>
```

```
</ErrorResponse>
```

You'll receive the same type of invalid region response from AWS products that are available in multiple regions if you submit requests to a region that differs from the region specified in your credential scope.

The credential must also specify the correct region for the service and action in your request.

The date that you use as part of the credential must match the date value in the `x-amz-date` header. For example, the following `x-amz-date` header value does not match the date value used in the `Credential` parameter that follows it.

```
x-amz-date:"20120224T213559Z"
Credential=AKIAIOSFODNN7EXAMPLE/20120225/us-east-1/iam/aws4_request
```

If you use this pairing of `x-amz-date` header and credential, you'll receive the following error response:

```
<ErrorResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>SignatureDoesNotMatch</Code>
    <Message>Date in Credential scope does not match YYYYMMDD from ISO-8601 version of date
 from HTTP: '20120225' != '20120224', from '20120 224T213559Z'.</Message>
  </Error>
  <RequestId>9d6ddd2b-5f2f-11e1-b901-a702cd369eb8</RequestId>
</ErrorResponse>
```

An expired signature can also generate an error response. For example, the following error response was generated due to an expired signature.

```
<ErrorResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">
  <Error>
    <Type>Sender</Type>
    <Code>SignatureDoesNotMatch</Code>
    <Message>Signature expired: 20120306T074514Z is now earlier than 20120306T074556Z
 (20120306T080056Z - 15 min.)</Message>
  </Error>
  <RequestId>fcc88440-5dec-11e1-b901-a702cd369eb8</RequestId>
</ErrorResponse>
```

Troubleshooting AWS Signature Version 4 Key Signing Errors

Errors that are caused by an incorrect derivation of the signing key or improper use of cryptography are more difficult to troubleshoot. The error response will tell you that the signature does not match. If you verified that the canonical string and the string to sign are correct, the cause of the signature mismatch is most likely one of the two following issues:

- The secret access key does not match the access key ID that you specified in the `Credential` parameter.
- There is a problem with your key derivation code.

To check whether the secret key matches the access key ID, you can use your secret key and access key ID with a known working implementation. One way is to use one of the AWS SDKs to write a program that makes a simple request to AWS using the access key ID and secret access key that you want to use.

To check whether your key derivation code is correct, you can compare it to our example derivation code. For more information, see Examples of How to Derive a Signing Key for Signature Version 4 (p. 111).

Service-Specific Reference for Signature Version 4

To learn more about making and signing HTTP requests in the context of specific AWS services, see the documentation for the following services:

- Amazon API Gateway
- Amazon CloudSearch
- Amazon CloudWatch
- AWS Data Pipeline
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Elastic Transcoder
- Amazon Glacier
- Amazon Mobile Analytics
- Amazon Relational Database Service (Amazon RDS)
- Amazon Simple Email Service (Amazon SES)
- Amazon Simple Queue Service (Amazon SQS)
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Workflow Service (Amazon SWF)
- AWS WAF

Signature Version 2 Signing Process

You can use Signature Version 2 to sign API requests. However, we recommend that you sign your request with Signature Version 4. For more information, see Signature Version 4 Signing Process (p. 100).

Supported Regions and Services

The following regions don't support Signature Version 2. You must use Signature Version 4 to sign API requests in these regions:

- US East (Ohio) Region
- Canada (Central) Region
- Asia Pacific (Mumbai) Region
- Asia Pacific (Seoul) Region
- EU (Frankfurt) Region
- EU (London) Region
- China (Beijing) Region

The following services support Signature Version 2 in all other regions.

AWS services that support Signature Version 2

Auto Scaling	Auto Scaling API Reference
AWS CloudFormation	AWS CloudFormation API Reference
Amazon CloudWatch	Amazon CloudWatch API Reference
AWS Elastic Beanstalk	Elastic Beanstalk API Reference

Amazon Elastic Compute Cloud (Amazon EC2)	Amazon EC2 API Reference
Elastic Load Balancing	Elastic Load Balancing API Reference version 2012-06-01
Amazon EMR	Amazon EMR API Reference
Amazon ElastiCache	Amazon ElastiCache API Reference
AWS Identity and Access Management (IAM)	IAM API Reference
AWS Import/Export	AWS Import/Export API Reference
Amazon Relational Database Service (Amazon RDS	Amazon Relational Database Service API Reference
Amazon Simple Notification Service (Amazon SNS)	Amazon Simple Notification Service API Reference
Amazon Simple Queue Service (Amazon SQS)	Amazon Simple Queue Service API Reference
Amazon SimpleDB	Amazon SimpleDB API Reference

Components of a Query Request for Signature Version 2

AWS requires that each HTTP or HTTPS Query request formatted for Signature Version 2 contains the following:

Endpoint

Also known as the host part of an HTTP request. This is the DNS name of the computer where you send the Query request. This is different for each AWS region. For the list of endpoints for each service, see AWS Regions and Endpoints (p. 2).

Action

The action you want a web service to perform. This value determines the parameters used in the request.

AWSAccessKeyId

A value distributed by AWS when you sign up for an AWS account.

SignatureMethod

The hash-based protocol used to calculate the signature. This can be either HMAC-SHA1 or HMAC-SHA256 for Signature Version 2.

SignatureVersion

The version of the AWS signature protocol.

Timestamp

The time at which you make the request. Include this in the Query request to help prevent third parties from intercepting your request.

Required and optional parameters

Each action has a set of required and optional parameters that define the API call.

Signature

The calculated value that ensures the signature is valid and has not been tampered.

The following is an example Amazon EMR Query request formatted as an HTTPS GET request.

- The endpoint, `elasticmapreduce.amazonaws.com`, is the default endpoint and maps to the region us-east-1.
- The action is `DescribeJobFlows`, which requests information about one or more job flows.

Note

In the actual Query request, there are no spaces or newline characters. The request is a continuous line of text. The version below is formatted for human readability.

```
https://elasticmapreduce.amazonaws.com?
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Action=DescribeJobFlows
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2011-10-03T15%3A19%3A30
&Version=2009-03-31
&Signature=calculated value
```

How to Generate a Signature Version 2 for a Query Request

Web service requests are sent across the Internet and are vulnerable to tampering. To check that the request has not been altered, AWS calculates the signature to determine if any of the parameters or parameter values were changed en route. AWS requires a signature as part of every request.

Note

Be sure to URI encode the request. For example, blank spaces in your request should be encoded as %20. Although an unencoded space is normally allowed by the HTTP protocol specification, unencoded characters create an invalid signature in your Query request. Do *not* encode spaces as a plus sign (+) as this will cause errors.

The following topics describe the steps needed to calculate a signature using AWS Signature Version 2.

Task 1: Format the Query Request

Before you can sign the Query request, format the request in a standardized (canonical) format. This is needed because the different ways to format a Query request will result in different HMAC signatures. Format the request in a canonical format before signing. This ensures your application and AWS will calculate the same signature for a request.

To create the string to sign, you concatenate the Query request components. The following example generates the string to sign for the following call to the Amazon EMR API.

```
https://elasticmapreduce.amazonaws.com?
Action=DescribeJobFlows
&Version=2009-03-31
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2011-10-03T15:19:30
```

Note

In the preceding request, the last four parameters (AWSAccessKeyID through Timestamp) are called authentication parameters. They're required in every Signature Version 2 request. AWS uses them to identify who is sending the request and whether to grant the requested access.

To create the string to sign (Signature Version 2)

1. Start with the request method (either GET or POST), followed by a newline character. For human readability, the newline character is represented as `\n`.

   ```
   GET\n
   ```

2. Add the HTTP host header (endpoint) in lowercase, followed by a newline character. The port information is omitted if it is the standard port for the protocol (port 80 for HTTP and port 443 for HTTPS), but included if it is a nonstandard port.

   ```
   elasticmapreduce.amazonaws.com\n
   ```

3. Add the URL-encoded version of each path segment of the URI, which is everything between the HTTP host header to the question mark character (`?`) that begins the query string parameters, followed by a newline character. Don't encode the forward slash (`/`) that delimits each path segment.

   In this example, if the absolute path is empty, use a forward slash (`/`).

   ```
   /\n
   ```

4. a. Add the query string components, as UTF-8 characters which are URL encoded (hexadecimal characters must be uppercase). You do not encode the initial question mark character (?) in the request. For more information, see RFC 3986.

   b. Sort the query string components by byte order. Byte ordering is case sensitive. AWS sorts these components based on the raw bytes.

   For example, this is the original order for the query string components.

   ```
   Action=DescribeJobFlows
   Version=2009-03-31
   AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
   SignatureVersion=2
   SignatureMethod=HmacSHA256
   Timestamp=2011-10-03T15%3A19%3A30
   ```

   The query string components would be reorganized as the following:

   ```
   AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
   Action=DescribeJobFlows
   SignatureMethod=HmacSHA256
   SignatureVersion=2
   Timestamp=2011-10-03T15%3A19%3A30
   Version=2009-03-31
   ```

   c. Separate parameter names from their values with the equal sign character (=), even if the value is empty. Separate parameter and value pairs with the ampersand character (&). Concatenate the parameters and their values to make one long string with no spaces. Spaces within a parameter value are allowed, but must be URL encoded as %20. In the concatenated string, period characters (.) are not escaped. RFC 3986 considers the period character an unreserved character, so it is not URL encoded.

Note

RFC 3986 does not specify what happens with ASCII control characters, extended UTF-8 characters, and other characters reserved by RFC 1738. Since any values may be passed into a string value, these other characters should be percent encoded as %XY where X and Y are uppercase hex characters. Extended UTF-8 characters take the form %XY%ZA... (this handles multibytes).

The following example shows the query string components, with the parameters concatenated with the ampersand character (&), and sorted by byte order.

```
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Action=DescribeJobFlows&SignatureMethod=HmacSHA256&SignatureVer
```

5. To construct the finished canonical request, combine all the components from each step. As shown, each component ends with a newline character.

```
GET\n
elasticmapreduce.amazonaws.com\n
/\n
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Action=DescribeJobFlows&SignatureMethod=HmacSHA256&SignatureVer
```

Task 2: Calculate the Signature

After you've created the canonical string as described in Task 1: Format the Query Request (p. 128), calculate the signature by creating a hash-based message authentication code (HMAC) that uses either the HMAC-SHA1 or HMAC-SHA256 protocols. The HMAC-SHA256 is preferred.

In this example, the signature is calculated with the following canonical string and secret key as inputs to a keyed hash function:

- Canonical query string:

```
GET\n
elasticmapreduce.amazonaws.com\n
/\n
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Action=DescribeJobFlows&SignatureMethod=HmacSHA256&SignatureVersi
```

- Sample secret key:

```
wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

The resulting signature must be base-64 encoded.

```
i91nKc4PWAt0JJIdXwz9HxZCJDdiy6cf%2FMj6vPxyYIs%3D
```

Add the resulting value to the query request as a `Signature` parameter. When you add this parameter to the request, you must URI encode it just like any other parameter. You can use the signed request in an HTTP or HTTPS call.

```

https://elasticmapreduce.amazonaws.com?
AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Action=DescribeJobFlows&SignatureMethod=HmacSHA256&SignatureVersion
%2FMj6vPxyYIs%3D
```

Note

You can use temporary security credentials provided by AWS Security Token Service (AWS STS) to sign a request. The process is the same as using long-term credentials, but requests require an additional parameter for the security token.

The following request uses a temporary access key ID and the `SecurityToken` parameter.

For more information, see the following resources:

- The Amazon EMR Developer Guide has information about Amazon EMR API calls.
- The API documentation for each service has information about requirements and specific parameters for an action.
- The AWS SDKs offer functions to generate Query request signatures. To see an example using the AWS SDK for Java, see Using the Java SDK to Sign a Query Request (p. 132).

Troubleshooting Request Signatures Version 2

This section describes some error codes you might see when you are initially developing code to generate the signature to sign Query requests.

SignatureDoesNotMatch Signing Error in a web service

The following error response is returned when a web service attempts to validate the request signature by recalculating the signature value and generates a value that does not match the signature you appended to the request. This can occur because the request was altered between the time you sent it and the time it reached a web service endpoint (which is what the signature is designed to detect) or because the signature was calculated improperly. A common cause of the following error message is not properly creating the string to sign, such as forgetting to URL-encode characters such as the colon (:) and the forward slash (/) in Amazon S3 bucket names.

```
<ErrorResponse xmlns="http://elasticmapreduce.amazonaws.com/doc/2009-03-31">
  <Error>
    <Type>Sender</Type>
    <Code>SignatureDoesNotMatch</Code>
    <Message>The request signature we calculated does not match the signature you
 provided.
    Check your AWS Secret Access Key and signing method.
    Consult the service documentation for details.</Message>
  </Error>
  <RequestId>7589637b-e4b0-11e0-95d9-639f87241c66</RequestId>
</ErrorResponse>
```

IncompleteSignature Signing Error in a web service

The following error indicates that signature is missing information or has been improperly formed.

```
<ErrorResponse xmlns="http://elasticmapreduce.amazonaws.com/doc/2009-03-31">
  <Error>
    <Type>Sender</Type>
    <Code>IncompleteSignature</Code>
    <Message>Request must contain a signature that conforms to AWS standards</Message>
  </Error>
  <RequestId>7146d0dd-e48e-11e0-a276-bd10ea0cbb74</RequestId>
</ErrorResponse>
```

Using the Java SDK to Sign a Query Request

The following example uses the amazon.webservices.common package of the AWS SDK for Java to generate an AWS Signature Version 2 Query request signature. To do so, it creates an RFC 2104-compliant HMAC signature. For more information about HMAC, see HMAC: Keyed-Hashing for Message Authentication.

Note
Java is used as an example implementation. You can use the programming language of your choice to implement the HMAC algorithm to sign Query requests.

```
import java.security.SignatureException;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import com.amazonaws.util.*;

/**
* This class defines common routines for generating
* authentication signatures for AWS Platform requests.
*/
public class Signature {
    private static final String HMAC_SHA256_ALGORITHM = "HmacSHA256";

    /**
     * Computes RFC 2104-compliant HMAC signature.
     * * @param data
     * The signed data.
     * @param key
     * The signing key.
     * @return
     * The Base64-encoded RFC 2104-compliant HMAC signature.
     * @throws
     * java.security.SignatureException when signature generation fails
     */
    public static String calculateRFC2104HMAC(String data, String key)
    throws java.security.SignatureException
    {
        String result;
        try {

            // Get an hmac_sha256 key from the raw key bytes.
            SecretKeySpec signingKey = new SecretKeySpec(key.getBytes("UTF8"),
 HMAC_SHA256_ALGORITHM);

            // Get an hmac_sha256 Mac instance and initialize with the signing key.
            Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
            mac.init(signingKey);

            // Compute the hmac on input data bytes.
            byte[] rawHmac = mac.doFinal(data.getBytes("UTF8"));

            // Base64-encode the hmac by using the utility in the SDK
            result = BinaryUtils.toBase64(rawHmac);


        } catch (Exception e) {
            throw new SignatureException("Failed to generate HMAC : " + e.getMessage());
        }
        return result;
    }
}
```

AWS Service Limits

The following tables provide the default limits for AWS services for an AWS account. Unless otherwise noted, each limit is region-specific. Many services contain limits that cannot be changed. For more information about the limits for a specific service, see the documentation for that service.

If your support plan includes Trusted Advisor, you can use it to display your usage and limits for each service in a specific region. For more information, see Trusted Advisor.

You can take the following steps to request an increase for limits. These increases are not granted immediately, so it may take a couple of days for your increase to become effective.

To request a limit increase

1. Open the AWS Support Center page, sign in, if necessary, and then choose **Create Case**.
2. Under **Regarding**, choose **Service Limit Increase**.
3. Under **Limit Type**, choose the type of limit to increase, fill in the necessary fields in the form, and then choose your preferred method of contact.

Default Limits

Amazon API Gateway Limits

The following limits apply to configuring and running an API in Amazon API Gateway and can be increased upon request to optimize performances of a deployed API in Amazon API Gateway.

Resource or Operation	Default Limit
Throttle rate per account	1000 request per second (rps) with a burst limit of 2000 rps.
APIs per account	60
API keys per account	500
Custom authorizers per API	10
Client certificates per account	60
Documentation parts per API	2000
Resources per API	300
Stages per API	10
Usage plans per account	300
Usage plans per API key	10

All of the per API limits can only be increased on specific APIs.

For information about additional documented limits, see Limits in Amazon API Gateway in the *API Gateway Developer Guide*.

AWS Application Discovery Service Limits

Resource	Default Limit
Inactive agents heartbeating but not collecting data	10,000
Active agents sending data to the service	250
Total collected data for all agents, per day	10 GB
Data storage duration before being purged	90 days

Amazon AppStream Limits

Important
This information applies only to an older version of Amazon AppStream.

An Amazon AppStream account has a service limit of up to five concurrent streaming sessions:

- Up to two concurrent streaming application deployments using the interactive wizard.
- Up to three streaming applications in the **Building**, **Active**, or **Error** states.

For more information, see Amazon AppStream Application Lifecycle in the *Amazon AppStream Developer Guide*.

Amazon AppStream 2.0 Limits

Important
This information applies only to the latest version, Amazon AppStream 2.0.

Default Limits Per Region

Resource	Default Limit
Stacks	5 per account
Fleets	5 per account
Streaming instances	5 per account
Images	5 per account
Image builders	5 per account
Users	5 per account

Application Auto Scaling Limits

Resource	Default Limit
Scalable targets	500
Scaling policies per scalable target	50
Step adjustments per scaling policy	20

Amazon Athena Limits

Resource	Default Limit
Number of concurrent queries	5
Query timeout	30 minutes

Resource	Default Limit
Number of databases	100
Number of tables per database	100
Number of partitions per table	20,000

Auto Scaling Limits

Resource	Default Limit
Launch configurations	100
Auto Scaling groups	20
Scaling policies per Auto Scaling group	50
Scheduled actions per Auto Scaling group	125
Lifecycle hooks per Auto Scaling group	50
SNS topics per Auto Scaling group	10
Load balancers per Auto Scaling group	50
Target groups per Auto Scaling group	50
Step adjustments per scaling policy	20

For information about additional documented limits, see Auto Scaling Limits in the *Auto Scaling User Guide*.

AWS Batch Limits

Item	Default Limit
Maximum number of compute environments	10
Maximum number of job queues	5
Maximum number of compute environments per job queue	3

For more information about these limits, see Service Limits in the *AWS Batch User Guide*.

AWS Certificate Manager (ACM) Limits

Item	Default Limit
Number of ACM-provided certificates	100
Number of imported certificates	100

Item	Default Limit
Number of domain names per ACM-provided certificate	10

For more information about these limits, see Limits in the *AWS Certificate Manager User Guide*.

AWS CloudFormation Limits

Resource	Default Limit
Stacks	200

For information about additional documented limits, see AWS CloudFormation Limits in the *AWS CloudFormation User Guide*.

Amazon CloudFront Limits

Resource	Default Limit
Data transfer rate per distribution	40 Gbps
Requests per second per distribution	100,000
Web distributions per account	200
RTMP distributions per account	100
Alternate domain names (CNAMEs) per distribution	100
Origins per distribution	25
Cache behaviors per distribution	25
Whitelisted headers per cache behavior	10
Whitelisted cookies per cache behavior	10
SSL certificates per account when serving HTTPS requests using dedicated IP addresses (no limit when serving HTTPS requests using SNI)	2
Custom headers that you can have Amazon CloudFront forward to the origin	10 name–value

For information about additional documented limits, see Limits in the *Amazon CloudFront Developer Guide*.

AWS CloudHSM Limits

Resource	Default Limit
HSM appliances	3

Resource	Default Limit
High-availability partition groups	20
Clients	800

Amazon CloudSearch Limits

Resource	Default Limit
Partitions	10
Search instances	50

For information about additional documented limits, see Understanding Amazon CloudSearch Limits in the *Amazon CloudSearch Developer Guide*.

Amazon CloudWatch Limits

Resource	Default Limit	Comments
DescribeAlarms	3 transactions per second (TPS)	The maximum number of operation requests you can make per second without being throttled. You can request a limit increase.
GetMetricStatistics	400 transactions per second (TPS)	The maximum number of operation requests you can make per second without being throttled. You can request a limit increase.
ListMetrics	25 transactions per second (TPS)	The maximum number of operation requests you can make per second without being throttled. You can request a limit increase.
PutMetricAlarm	3 transactions per second (TPS)	The maximum number of operation requests you can make per second without being throttled. You can request a limit increase.
PutMetricData	150 transactions per second (TPS)	The maximum number of operation requests you can make per second without being throttled.

Resource	Default Limit	Comments
		You can request a limit increase.

For information about additional documented limits, see CloudWatch Limits in the *Amazon CloudWatch User Guide*.

Amazon CloudWatch Events Limits

Resource	Default Limit	Comments
Rules	50 per region per account	You can request a limit increase. Before requesting a limit increase, examine your rules. You may have multiple rules each matching to very specific events. Consider broadening their scope by using fewer identifiers in your Events and Event Patterns. In addition, a rule can invoke several targets each time it matches an event. Consider adding more targets to your rules.

For information about additional documented limits, see CloudWatch Events Limits in the *Amazon CloudWatch Events User Guide*.

Amazon CloudWatch Logs Limits

Resource	Default Limit	Comments
CreateLogGroup	500 log groups/account/region	If you exceed your log group limit, you get a `ResourceLimitExceeded` exception. You can request a limit increase.
DescribeLogStreams	5 transactions per second (TPS)/account/region	If you experience frequent throttling, you can request a limit increase.
FilterLogEvents	5 transactions per second (TPS)/account/region	This limit can be changed only in special circumstances. If you experience frequent throttling, contact AWS Support.
GetLogEvents	10 transactions per second (TPS)/account/region	We recommend subscriptions if you are continuously processing new data. If you need historical

Resource	Default Limit	Comments
		data, we recommend exporting your data to Amazon S3. This limit can be changed only in special circumstances. If you experience frequent throttling, contact AWS Support.

For information about additional documented limits, see CloudWatch Logs Limits in the *Amazon CloudWatch Logs User Guide*.

AWS CodeBuild Limits

Resource	Default Limit
Maximum number of build projects	1,000
Maximum number of concurrent running builds	20

For information about additional documented limits, see Limits for AWS CodeBuild in the *AWS CodeBuild User Guide*.

AWS CodeCommit Limits

Resource	Default Limit
Number of repositories	1,000 per AWS account

For information about additional documented limits, see Limits in AWS CodeCommit in the *AWS CodeCommit User Guide*.

AWS CodeDeploy Limits

Resource	Default Limit
Number of applications under an account in a single region	40
Number of concurrent deployments under an account	10
Number of deployment groups associated with a single application	50
Number of instances in a single deployment	50

For information about additional documented limits, see Limits in AWS CodeDeploy in the *AWS CodeDeploy User Guide*.

AWS CodePipeline Limits

Resource	Default Limit
Number of pipelines per AWS account	20
Number of stages in a pipeline	Minimum of 2, maximum of 10
Number of actions in a stage	Minimum of 1, maximum of 20
Number of parallel actions in a stage	5
Number of sequential actions in a stage	5
Number of custom actions per AWS account	50
Maximum number of revisions running across all pipelines	Five times the number of pipelines in your account
Maximum size of source artifacts	500 megabytes (MB)
Maximum number of times an action can be run per month	1,000 per calendar month

It may take up to two weeks to process requests for a limit increase.

For information about additional documented limits, see Limits in AWS CodePipeline in the *AWS CodePipeline User Guide*.

Amazon Cognito Your User Pools Limits

Resource	Default Limit
Maximum number of apps per user pool	25
Maximum number of user pools per account	60
Maximum number of user import jobs per user pool	50

For information about additional documented limits, see Limits in Amazon Cognito in the *Amazon Cognito Developer Guide*.

Amazon Cognito Federated Identities Limits

Resource	Default Limit
Maximum number of identity pools per account	60

For information about additional documented limits, see Limits in Amazon Cognito in the *Amazon Cognito Developer Guide*.

Amazon Cognito Sync Limits

Resource	Default Limit
Maximum number of datasets per identity	20
Maximum number of records per dataset	1024
Maximum size of a single dataset	1 MB

For information about additional documented limits, see Limits in Amazon Cognito in the *Amazon Cognito Developer Guide*.

AWS Config Limits

Resource	Default Limit	Notes
Number of AWS Config rules per region in your account	50	You can request a limit increase.

AWS Data Pipeline Limits

Attribute	Limit	Adjustable
Number of pipelines	100	Yes
Number of objects per pipeline	100	Yes
Number of active instances per object	5	Yes
Number of fields per object	50	No
Number of UTF8 bytes per field name or identifier	256	No
Number of UTF8 bytes per field	10,240	No
Number of UTF8 bytes per object	15,360 (including field names)	No
Rate of creation of an instance from an object	1 per 5 minutes	No
Retries of a pipeline activity	5 per task	No
Minimum delay between retry attempts	2 minutes	No
Minimum scheduling interval	15 minutes	No
Maximum number of roll-ups into a single object	32	No
Maximum number of EC2 instances per Ec2Resource object	1	No

For additional limits, see AWS Data Pipeline Limits in the *AWS Data Pipeline Developer Guide*.

AWS Database Migration Service Limits

Resource	Default Limit
Replication instances	20
Total amount of storage	6 TB
Replication subnet groups	20
Subnets per replication subnet group	20
Endpoints	100
Tasks	200
Endpoints per instance	20

AWS Device Farm Limits

Resource	Default Limit	Comments
App file size you can upload	4 GB	
Number of devices that AWS Device Farm can test during a run	5	This limit can be increased to 100 upon request.
Number of devices you can include in a test run	None	
Number of runs you can schedule	None	
Duration of a remote access session	60 minutes	

AWS Direct Connect Limits

Resource	Default Limit	Comment
Virtual interfaces per AWS Direct Connect connection	50	To increase this limit, submit a request.
Active AWS Direct Connect connections per region per account	10	To increase this limit, submit a request.
Routes per Border Gateway Protocol (BGP) session	100	This limit cannot be increased.
Connections per link aggregation group (LAG)	4	To increase this limit, submit a request.
Link aggregation groups (LAGs) per region	10	To increase this limit, submit a request.

AWS Directory Service Limits

Resource	Default Limit
AD Connector directories	10
AWS Directory Service for Microsoft Active Directory (Enterprise Edition) directories	10
Simple AD directories	10
Manual snapshots	5 per Microsoft AD
Manual snapshots	5 per Simple AD

For information about additional documented limits, including limits on Amazon Cloud Directory, see AWS Directory Service Limits in the *AWS Directory Service Admin Guide*.

Amazon DynamoDB Limits

Resource	Default Limit
US East (N. Virginia) Region: Maximum capacity units per table or global secondary index	40,000 read capacity units and 40,000 write capacity units
US East (N. Virginia) Region: Maximum capacity units per account	80,000 read capacity units and 80,000 write capacity units
All other regions: Maximum capacity units per table or global secondary index	10,000 read capacity units and 10,000 write capacity units
All other regions: Maximum capacity units per account	20,000 read capacity units and 20,000 write capacity units
Maximum number of tables	256

For information about additional documented limits, see Limits in Amazon DynamoDB in the *Amazon DynamoDB Developer Guide*.

Amazon EC2 Container Registry (Amazon ECR) Limits

Resource	Default Limit
Maximum number of repositories per account	1,000
Maximum number of images per repository	1,000

For information about additional documented limits, see Amazon ECR Service Limits in the *Amazon EC2 Container Registry User Guide*.

Amazon EC2 Container Service (Amazon ECS) Limits

Resource	Default Limit
Number of clusters per region per account	1000
Number of container instances per cluster	1000
Number of services per cluster	500

For information about additional documented limits, see Amazon ECS Service Limits in the *Amazon EC2 Container Service Developer Guide*.

Amazon EC2 Systems Manager Limits

Resource	Default Limit
Managed instances	500 Each AWS account can register/activate a maximum of 500 managed instances in a region.
Systems Manager documents	200 Each AWS account can create a maximum of 200 documents per region.
Privately shared Systems Manager document	20 A single Systems Manager document can be shared with a maximum of 20 AWS accounts.
Publicly shared Systems Manager document	5 Each AWS account can publicly share a maximum of five documents.
Document associations	10,000 Each Systems Manager document can be associated with a maximum of 10,000 instances.
Inventory data collected per instance per call	1 MB

Resource	Default Limit
	This maximum adequately supports most inventory collection scenarios. When this limit is reached, no new inventory data is collected for the instance. Inventory data previously collected is stored until the expiration.
Inventory data collected per instance per day	5 MB When this limit is reached, no new inventory data is collected for the instance. Inventory data previously collected is stored until the expiration.
Custom Inventory Types	20 You can add up to 20 custom inventory types.
Custom Inventory Type Size	4 KB This is the maximum size of the type, not the inventory collected.
Custom Inventory Type Attributes	50 This is the maximum number of attributes within the custom inventory type.
Inventory data expiration	30 days If you terminate an instance, inventory data for that instance is deleted immediately. For running instances, inventory data older than 30 days is deleted. If you need to store inventory data longer than 30 days, you can use AWS Config to record history or periodically query and upload the data to an Amazon S3 bucket. For more information, see, Recording Amazon EC2 managed instance inventory in the *AWS Config Developer Guide*.
Maintenance Windows per account	50
Tasks per Maintenance Window	20
Targets per Maintenance Window	50
Instance IDs per target	50

Resource	Default Limit
Targets per task	10
Concurrent executions of a single Maintenance Window	1
Concurrent executions of Maintenance Windows	5
Maintenance Window execution history retention	30 days
Maximum number of parameters per account	1000
Max size for parameter value	4096 characters
Max history for a parameter	100 past values
Patch baselines per account	25
Patch groups per patch baseline	25

AWS Elastic Beanstalk Limits

Resource	Default Limit
Applications	75
Application Versions	1000
Environments	200

Amazon Elastic Block Store (Amazon EBS) Limits

Resource	Default Limit
Number of EBS volumes	5,000
Number of EBS snapshots	10,000
Total volume storage of General Purpose SSD (`gp2`) volumes	20 TiB
Total volume storage of Provisioned IOPS SSD (`io1`) volumes	20 TiB
Total volume storage of Throughput Optimized HDD (`st1`)	20 TiB
Total volume storage of Cold HDD (`sc1`)	20 TiB
Total volume storage of Magnetic volumes	20 TiB
Total provisioned IOPS	40,000

For information about additional documented limits, see Amazon EC2 Service Limits in the *Amazon EC2 User Guide for Linux Instances*.

Amazon Elastic Compute Cloud (Amazon EC2) Limits

Resource	Default Limit
Elastic IP addresses for EC2-Classic	5
Security groups for EC2-Classic per instance	500
Rules per security group for EC2-Classic	100
Key pairs	5,000
Throttle on the emails that can be sent from your Amazon EC2 account	Throttle applied
On-Demand Instances	Limits vary depending on instance type. For more information, see How many instances can I run in Amazon EC2.
Spot Instances	Limits vary depending on instance type, region, and account. For more information, see Spot Instance Limits.
Reserved Instances	20 instance reservations per Availability Zone, per month.
Dedicated Hosts	Up to two Dedicated Hosts per instance family, per region can be allocated.
AMI Copies	Destination regions are limited to 50 concurrent AMI copies at a time, with no more than 25 of those coming from a single source region.

For information about related limits for EC2-VPC, see Amazon Virtual Private Cloud (Amazon VPC) Limits (p. 168).

For information about viewing your current limits, see Amazon EC2 Service Limits in the *Amazon EC2 User Guide for Linux Instances*.

Amazon Elastic File System Limits

Resource	Default Limit
Total throughput per file system	3 GB/s for all connected clients

For information about additional documented limits, see Amazon EFS Limits in the *Amazon Elastic File System User Guide*.

Elastic Load Balancing Limits

Elastic Load Balancing supports two types of load balancers: Application Load Balancers and Classic Load Balancers.

Application Load Balancers

Resource	Default Limit
Load balancers per region	20 †
Target groups per region	200
Listeners per load balancer	10
Targets per load balancer	1000
Subnets per Availability Zone per load balancer	1
Security groups per load balancer	5
Rules per load balancer (not counting default rules)	10
Number of times a target can be registered per load balancer	100
Load balancers per target group	1
Targets per target group	1000

Classic Load Balancers

Resource	Default Limit
Load balancers per region	20 †
Listeners per load balancer	100
Security groups per load balancer	5
Subnets per Availability Zone per load balancer	1

† This limit includes both your Application LoadBalancers and your Classic Load Balancers. This limit can be increased upon request.

Amazon Elastic Transcoder Limits

Resource	Default Limit
Pipelines per region	4
User-defined presets	50
Maximum number of jobs processed simultaneously by each pipeline	US East (N. Virginia) Region – 20 US West (N. California) Region – 12 US West (Oregon) Region – 20

Resource	Default Limit
	Asia Pacific (Mumbai) Region – 12 Asia Pacific (Singapore) Region – 12 Asia Pacific (Sydney) Region – 12 Asia Pacific (Tokyo) Region – 12 EU (Ireland) Region – 20

It may take up to two weeks to process requests for a limit increase.

For information about additional documented limits, see Amazon Elastic Transcoder limits in the *Amazon Elastic Transcoder Developer Guide*.

Amazon ElastiCache Limits

Resource	Default Limit	Description
Nodes per region	100	The maximum number of nodes across all clusters in a region.
Nodes per cluster (Memcached)	20	The maximum number of nodes in an individual Memcached cluster.
Nodes per cluster (Redis)	1	The maximum number of nodes in an individual Redis cluster.
Clusters per replication group (Redis)	6	The maximum number of clusters in a Redis replication group. One is the read/write primary. All others are read-only replicas.
Parameter groups per region	20	The maximum number of parameters groups you can create in a region.
Security groups per region	50	The maximum number of security groups you can create in a region.
Subnet groups per region	50	The maximum number of subnet groups you can create in a region.
Subnets per subnet group	20	The maximum number of subnets you can define for a subnet group.

These limits are global limits per customer account. To exceed these limits, make your request using the ElastiCache Node request form.

Amazon Elasticsearch Service Limits

Resource	Default Limit
Number of Amazon ES instances per cluster	20

Amazon GameLift Limits

Resource	Default Limit
Aliases	20
Fleets	20
Builds	1000
Total size of builds	100 GB
Log upload size per game session	200 MB
On-demand instances	Limits vary depending on instance type; 20 instances per account, regardless of instance type
Server processes per instance	1 with GameLift SDK v2.x 50 with GameLift SDK v3.x and up
Player sessions per game session	200

For information about additional documented limits, see Scaling Amazon Elastic Compute Cloud (Amazon EC2) Instances in the *Amazon GameLift Developer Guide*.

AWS Identity and Access Management (IAM) Limits

Resource	Default Limit
Groups per account	100
Instance profiles	100
Roles	250
Server certificates	20
Users	5000

For information about additional documented limits, see Limitations on IAM Entities and Objects in the *IAM User Guide*.

AWS Import/Export Limits

AWS Snowball (Snowball)

Resource	Default Limit	Comments
Snowball	1	To increase this limit, contact AWS Support.

Amazon Inspector Limits

Resource	Default Limit
Running agents	500
Assessment runs	50,000
Assessment templates	500
Assessment targets	50

For more information, see the Amazon Inspector User Guide.

AWS IoT Limits

Message Broker Limits

Client ID size	128 bytes of UTF-8 encoded characters.
Connection inactivity (keep-alive interval)	By default, an MQTT client connection is disconnected after 30 minutes of inactivity. When the client sends a PUBLISH, SUBSCRIBE, PING, or PUBACK message, the inactivity timer is reset. A client can request a shorter keep-alive interval by specifying a value between 5-1,200 seconds in the MQTT CONNECT message sent to the server. If a keep-alive value is specified, the server disconnects the client if it does not receive a PUBLISH, SUBSCRIBE, PINGREQ, or PUBACK message within a period 1.5 times the requested interval. The keep-alive timer starts after the sender sends a CONNACK. If a client sends a keep-alive value of zero, the default keep-alive behavior remains in place. If a client requests a keep-alive shorter than 5 seconds, the server treats the client as though it requested a keep-alive interval of 5 seconds.

	The keep-alive timer begins immediately after the server returns a CONNACK to the client. There might be a brief delay between the client's sending of a CONNECT message and the start of keep-alive behavior.
Connect requests per second per account	AWS IoT limits an account to a maximum of 300 MQTT CONNECT requests per second.
Maximum number of slashes in topic and topic filter	A topic provided while publishing a message or a topic filter provided while subscribing can have no more than 7 forward slashes (/).
Maximum inbound unacknowledged messages	The message broker allows 100 in-progress unacknowledged messages per client. (This limit is applied across all messages that require ACK.) When this limit is reached, no new messages are accepted from this client until an ACK is returned by the server.
Maximum outbound unacknowledged messages	The message broker allows only 100 in-progress unacknowledged messages per client. (This limit is applied across all messages that require ACK.) When this limit is reached, no new messages are sent to the client until the client acknowledges the in-progress messages.
Maximum retry interval for delivering QoS 1 messages	If a connected client is unable to receive an ACK on a QoS 1 message for one hour, the message broker drops the message. The client might be unable to receive the message if it has 100 in-flight messages, it is being throttled due to large payloads, or other errors.
Maximum subscriptions per subscribe call	A single SUBSCRIBE call is limited to request a maximum of eight subscriptions.
Message size	The payload for every PUBLISH message is limited to 128 KB. The AWS IoT service rejects messages larger than this size.

Publish requests per second per account	9000 per second per account (inbound publishes - max. 3000 per second, outbound publishes - max. 6000 per second) Inbound publishes count for all the messages that the message broker processes before routing the messages to the subscribed clients or the rules engine. For example, a single message published on `$aws/things/device/shadow/update` topic can result in publishing three additional messages to `$aws/things/device/shadow/update/accepted`, `$aws/things/device/shadow/update/documents`, `$aws/things/device/shadow/delta` topics. In this case, AWS IoT counts those as 4 inbound publishes towards this limit. However, a single message to an unreserved topic like `"a/b"` is counted only as a single inbound publish Outbound publishes count for every message that resulted in matching a client's subscription or matching a rules engine subscription. For example, two clients are subscribed to topic filter `'a/b'` and a rule is subscribed to topic filter `'a/#'`. An inbound publish message on topic 'a/b' results in a total of 3 outbound publishes. **Note** Inbound and outbound publishes cannot be traded for each other, for example, if only 1,000 inbound publishes per second are used, the maximum outbound publishes per second remains 6,000.
Restricted client ID prefix	'$' is reserved for internally generated client IDs.
Restricted topic prefix	Topics beginning with '$' are considered reserved and are not supported for publishing and subscribing except when working with the Thing Shadows service.
Subscriptions per second per account	AWS IoT limits an account to a maximum of 500 subscriptions per second. For example, if there are two MQTT SUBSCRIBE calls within a second with 3 subscriptions (topic filters) each, AWS IoT counts those as 6 subscriptions towards this limit.
Subscriptions per session	The message broker limits each client session to subscribe to up to 50 subscriptions. A SUBSCRIBE request that pushes the total number of subscriptions past 50 results in the connection being disconnected.
Thing name size	128 bytes of UTF-8 encoded characters. This limit applies for both the thing registry and Thing Shadow services.

Throughput per connection	AWS IoT limits the ingress and egress rate on each client connection to 512 KB/s. Data sent or received at a higher rate is throttled to this throughput.
Topic size	The topic passed to the message broker when publishing a message cannot exceed 256 bytes of UTF-8 encoded characters.
WebSocket connection duration	WebSocket connections are limited to 24 hours. If the limit is exceeded, the WebSocket connection is automatically closed when an attempt is made to send a message by the client or server. To maintain an active WebSocket connection for longer than 24 hours, simply close and reopen the WebSocket connection from the client side before the time limit elapses. AWS IoT supports keep-alive values specified in MQTT CONNECT messages. When a client specifies a keep-alive value, the client tells the server to disconnect the client and transmit any last-will message associated with the MQTT session if the server does not receive a message (PUBLISH, SUBSCRIBE, PUBACK, PINGREQ) within 1.5 times the keep-alive period. AWS IoT supports keep-alive values between 5 seconds and 20 minutes. If a client requests no keep-alive (that is, sets the field to 0 in the MQTT CONNECT message), the server sets the keep-alive value to 20 minutes, which corresponds to the maximum idle time supported by AWS IoT of 30 minutes. Most MQTT clients (including the AWS SDK clients) support keep-alive values by sending a PINGREQ if the keep-alive period expires without the transmission of any other message by the client.

Device Shadow Limits

Maximum depth of JSON device state documents	The maximum number of levels in the `desired` or `reported` section of the JSON device state document is 5. For example:

```
"desired": {
    "one": {
        "two": {
            "three": {
                "four": {
                    "five":{
                    }
                }
            }
        }
    }
```

	`}`
Maximum number of in-flight, unacknowledged messages	The Thing Shadows service supports up to 10 in-flight unacknowledged messages. When this limit is reached, all new shadow requests is rejected with a 429 error code.
Maximum number of JSON objects per AWS account	There is no limit on the number of JSON objects per AWS account.
Maximum size of a JSON state document	8 KB.
Maximum size of a thing name	128 bytes of UTF-8 encoded characters.
Shadow lifetime	A thing shadow is deleted by AWS IoT if it has not been updated or retrieved in more than one year.

Security and Identity Limits

Maximum number of CA certificates with the same subject field allowed per AWS account per region	10
Maximum number of policies that can be attached to a certificate or Amazon Cognito identity	10
Maximum number of named policy versions	5
Maximum policy document size	2048 characters (excluding white space)
Maximum number of device certificates that can be registered per second	15

Throttling Limits

API	Transaction per Second
AcceptCertificateTransfer	10
AttachPrincipalPolicy	15
AttachThingPrincipal	15
CancelCertificateTransfer	10
CreateCertificateFromCsr	15
CreatePolicy	10
CreatePolicyVersion	10
CreateThing	15
CreateThingType	15
DeleteCertificate	10

API	Transaction per Second
DeleteCACertificate	10
DeletePolicy	10
DeletePolicyVersion	10
DeleteThing	15
DeleteThingType	15
DeprecateThingType	15
DescribeCertificate	10
DescribeCACertificate	10
DescribeThing	10
DescribeThingType	10
DetachThingPrincipal	15
DetachPrincipalPolicy	15
DeleteRegistrationCode	10
GetPolicy	10
GetPolicyVersion	15
GetRegistrationCode	10
ListCACertificates	10
ListCertificates	10
ListCertificatesByCA	10
ListOutgoingCertificates	10
ListPolicies	10
ListPolicyPrincipals	10
ListPolicyVersions	10
ListPrincipalPolicies	15
ListPrincipalThings	10
ListThings	10
ListThingPrincipals	10
ListThingTypes	10
RegisterCertificate	10
RegisterCACertificate	10
RejectCertificateTransfer	10

API	Transaction per Second
SetDefaultPolicyVersion	10
TransferCertificate	10
UpdateCertificate	10
UpdateCACertificate	10
UpdateThing	10

AWS IoT Rules Engine Limits

Maximum number of rules per AWS account	1000
Actions per rule	A maximum of 10 actions can be defined per rule.
Rule size	Up to 256 KB of UTF-8 encoded characters (including white space).

AWS Key Management Service (AWS KMS) Limits

Resource	Default Limit
Customer Master Keys (CMKs)	1000
Aliases	1100
Grants per CMK	2500
Grants for a given principal per CMK	30
Requests per second	Varies by API operation; see Limits in the *AWS Key Management Service Developer Guide*.

All limits in the preceding table apply per region and per AWS account.

For information about additional documented limits, see Limits in the *AWS Key Management Service Developer Guide*.

Amazon Kinesis Firehose Limits

Resource	Default Limit
Delivery streams per region	20
Delivery stream capacity †	2,000 transactions/second 5,000 records/second 5 MB/second

† The three capacity limits scale proportionally. For example, if you increase the throughput limit to 10MB/second, the other limits increase to 4,000 transactions/second and 10,000 records/second.

For information about additional documented limits, see Amazon Kinesis Firehose Limits in the *Amazon Kinesis Firehose Developer Guide*.

Amazon Kinesis Streams Limits

Resource	Default Limit
Shards per region	US East (N. Virginia) Region – 50 US West (Oregon) Region – 50 EU (Ireland) Region – 50 All other supported regions – 25

For information about additional documented limits, see Amazon Kinesis Streams Limits in the *Amazon Kinesis Streams Developer Guide*.

AWS Lambda Limits

Resource	Limit
Concurrent requests safety throttle per account	100

For information about additional documented limits, see AWS Lambda Limits in the *AWS Lambda Developer Guide*.

Amazon Lightsail Limits

Resource	Default Limit	Comment
Number of instances	20 per account	This limit cannot be increased.
Number of Elastic IP addresses	5 per account	This limit cannot be increased.
Number of parallel SSH connections	3 x the number of instances in the account	This limit cannot be increased.
Number of hosted zones	3 per account	This limit cannot be increased.

Amazon Machine Learning (Amazon ML) Limits

Resource	Default Limit
Data file size*	100 GB

Resource	Default Limit
Batch prediction input size	1 TB
Batch prediction input (number of records)	100 million
Number of variables in a data file (schema)	1,000
Recipe complexity (number of processed output variables)	10,000
Transactions Per Second for each real-time prediction endpoint	200
Total Transactions Per Second for all real-time prediction endpoints	10,000
Total RAM for all real-time prediction endpoints	10 GB
Number of simultaneous jobs	5
Longest run time for any job	7 days
Number of classes for multiclass ML models	100
ML model size	2 GB

Note
The size of your data files is limited to ensure that jobs finish in a timely manner. Jobs that have been running for more than seven days are automatically terminated, resulting in a FAILED status.

For information about additional documented limits, see Amazon ML Limits in the *Amazon Machine Learning Developer Guide*.

AWS OpsWorks for Chef Automate Limits

Resource	Default Limit
Chef servers	5
User-initiated (manual) backup generations	10
Automated (scheduled) backup generations	30

AWS OpsWorks Stacks Limits

Resource	Default Limit
Stacks	40
Layers per stack	40
Instances per stack	40
Apps per stack	40

AWS Organizations Limits

Resource	Default Limit
Accounts per organization	20
Invitations sent per day	20

For information about additional documented limits, see Limits of AWS Organizations in the *AWS Organizations User Guide*.

Amazon Polly Limits

- Throttle rate per IP address: 100 transactions (requests) per second (tps) with a burst limit of 120 tps.
- Throttle rate per operation:

Throttle Rate per Operation

Operation	Limit
Lexicon	

`DeleteLexicon`
`PutLexicon`
`GetLexicon`
`ListLexicons`
	Any 2 transactions per second (tps) from these operations combined.
	Maximum allowed burst of 4 tps.

Speech	
`DescribeVoices`	80 rps with a burst limit of 100 tps
`SynthesizeSpeech`	80 rps with a burst limit of 100 tps

Amazon Pinpoint Limits

Resource	Default Limit
Active campaigns per account	100
Apps per account	100
Concurrent endpoint import jobs per account	2
Custom event types per app	1500
Endpoint custom attributes per app	40
Endpoints per mobile app user	10
Message sends per campaign activity	100 million

Resource	Default Limit
Segments per app	200
Total file size per endpoint import job	1 GB

Amazon Redshift Limits

Resource	Default Limit
Nodes per cluster	101
Nodes	200
Reserved Nodes	200
Snapshots	20
Parameter Groups	20
Security Groups	20
Subnet Groups	20
Subnets per Subnet Group	20
Event Subscriptions	20

For information about additional documented limits, see Limits in Amazon Redshift in the *Amazon Redshift Cluster Management Guide*.

Amazon Relational Database Service (Amazon RDS) Limits

Resource	Default Limit
Clusters	40
Cluster parameter groups	50
DB Instances	40
Event subscriptions	20
Manual snapshots	100
Manual cluster snapshots	50
Option groups	20
Parameter groups	50
Read replicas per master	5

Resource	Default Limit
Reserved instances (purchased per month)	40
Rules per security group	20
Security groups	25
Security groups (VPC)	5
Subnet groups	50
Subnets per subnet group	20
Tags per resource	50
Total storage for all DB instances	100 TB

Amazon Route 53 Limits

Resource	Default Limit
Hosted zones	500
Domains	50
Resource record sets per hosted zone	10,000
Reusable delegation sets	100
Hosted zones that can use the same reusable delegation set	100
Amazon VPCs that you can associate with a private hosted zone	100
Health checks	50
Traffic policies	50
Policy records	5

For information about additional documented limits, see Amazon Route 53 Limits in the *Amazon Route 53 Developer Guide*.

AWS Server Migration Service Limits

Resource	Default Limit
Concurrent VM migrations	50 per account
Maximum duration of service usage per VM (not per account), beginning with the initial replication of a VM. We terminate an ongoing replication after this period, unless a customer requests a limit increase.	90 days

AWS Service Catalog Limits

Resource	Default Limit
Portfolios	25 per account
Users, groups, and roles	25 per portfolio
Products	25 per portfolio, 25 total per account
Product versions	50 per product
Constraints	25 per product per portfolio
Tags	20 per product, 20 per portfolio, 50 per provisioned product
Stacks	200 (AWS CloudFormation limit)

AWS Shield Advanced Limits

AWS Shield Advanced offers advanced monitoring and protection for up to 100 CloudFront distributions, Amazon Route 53 hosted zones or Elastic Load Balancing resources combined.

Amazon Simple Email Service (Amazon SES) Limits

The following are the default limits for Amazon SES in the sandbox environment.

Resource	Default Limit
Daily sending quota	200 messages per 24-hour period.
Maximum send rate	1 email per second. **Note** The rate at which Amazon SES accepts your messages might be less than the maximum send rate.
Recipient address verification	All recipient addresses must be verified.

For information about additional documented limits, see Limits in Amazon SES in the *Amazon Simple Email Service Developer Guide*.

Amazon Simple Notification Service (Amazon SNS) Limits

Resource	Default Limit
Topics	100,000
Account spend threshold for SMS	1.00 USD
Delivery rate for promotional SMS messages	20 messages per second
Delivery rate for transactional SMS messages	20 messages per second

To increase any of these limits, submit a request.

Amazon SNS API Throttling Limits

API	Transaction per Second
ListEndpointsByPlatformApplication	30
ListTopics	30
ListPlatformApplications	15
ListSubscriptions	30
ListSubscriptionsByTopic	30
Subscribe	100
Unsubscribe	100

Amazon Simple Queue Service (Amazon SQS)

For information about additional documented limits, see Limits, Restrictions in the Amazon SQS FAQs and Amazon SQS Limits in the *Amazon Simple Queue Service Developer Guide*.

Amazon Simple Storage Service (Amazon S3) Limits

Resource	Default Limit
Buckets	100 per account

For information about additional documented limits, see Amazon S3 limits in the Amazon Simple Storage Service Developer Guide.

Amazon Simple Workflow Service (Amazon SWF) Limits

For information about additional documented limits, see Amazon SWF Service Limits in the *Amazon Simple Workflow Service Developer Guide*.

Amazon SimpleDB Limits

Resource	Default Limit
Domains	250

For information about additional documented limits, see Amazon SimpleDB Limits in the *Amazon SimpleDB Developer Guide*.

Amazon Virtual Private Cloud (Amazon VPC) Limits

Resource	Default limit	Comments
VPCs per region	5	The limit for Internet gateways per region is directly correlated to this one. Increasing this limit increases the limit on Internet gateways per region by the same amount. To increase this limit, submit a request.
Subnets per VPC	200	To increase this limit, submit a request.
Internet gateways per region	5	This limit is directly correlated with the limit on VPCs per region. You cannot increase this limit individually; the only way to increase this limit is to increase the limit on VPCs per region. Only one Internet gateway can be attached to a VPC at a time.
Egress-only Internet gateways per region	5	This limit is directly correlated with the limit on VPCs per region. You cannot increase this limit individually; the only way to increase this limit is to increase the limit on VPCs per region. Only one egress-only Internet gateway can be attached to a VPC at a time.
Virtual private gateways per region	5	To increase this limit, contact AWS Support; however, only one virtual private gateway can be attached to a VPC at a time.
Customer gateways per region	50	To increase this limit, contact AWS Support.
VPN connections per region	50	To increase this limit, submit a request.

Resource	Default limit	Comments
VPN connections per VPC (per virtual private gateway)	10	To increase this limit, submit a request.
Route tables per VPC	200	Including the main route table. You can associate one route table to one or more subnets in a VPC.
Routes per route table (non-propagated routes)	50	This is the limit for the number of non-propagated entries per route table. You can submit a request for an increase of up to a maximum of 100; however, network performance may be impacted. This limit is enforced separately for IPv4 routes and IPv6 routes (50 each, and a maximum of 100 each).
BGP advertised routes per route table (propagated routes)	100	You can have up to 100 propagated routes per route table; however, the total number of propagated and non-propagated entries per route table cannot exceed 100. For example, if you have 50 non-propagated entries (the default limit for this type of entry), you can only have 50 propagated entries. This limit cannot be increased. If you require more than 100 prefixes, advertise a default route.
Elastic IP addresses per region for each AWS account	5	This is the limit for the number of VPC Elastic IP addresses you can allocate within a region. This is a separate limit from the Amazon EC2 Elastic IP address limit. To increase this limit, submit a request.
Security groups per VPC	500	To increase this limit, you can submit a request.
Inbound or outbound rules per security group	50	You can have 50 inbound and 50 outbound rules per security group (giving a total of 100 combined inbound and outbound rules). To increase or decrease this limit, you can contact AWS Support — a limit change applies to both inbound and outbound rules. However, the multiple of the limit for inbound or outbound rules per security group and the limit for security groups per network interface cannot exceed 250. For example, if you increase the limit to 100, we decrease your number of security groups per network interface to 2. This limit is enforced separately for IPv4 rules and IPv6 rules. A rule that references a security group counts as one rule for IPv4 and one rule for IPv6.

Resource	Default limit	Comments
Security groups per network interface	5	To increase or decrease this limit, you can contact AWS Support. The maximum is 16. The multiple of the limit for security groups per network interface and the limit for rules per security group cannot exceed 250. For example, if you want 10 security groups per network interface, we decrease your number of rules per security group to 25.
Network interfaces per instance	-	This limit varies by instance type. For more information, see IP Addresses Per ENI Per Instance Type.
Network interfaces per region	350	This limit is the greater of either the default limit (350) or your On-Demand Instance limit multiplied by 5. The default limit for On-Demand Instances is 20. If your On-Demand Instance limit is below 70, the default limit of 350 applies. You can increase the number of network interfaces per region by contacting AWS Support, or by increasing your On-Demand Instance limit.
Network ACLs per VPC	200	You can associate one network ACL to one or more subnets in a VPC. This limit is not the same as the number of rules per network ACL.
Rules per network ACL	20	This is the one-way limit for a single network ACL, where the limit for ingress rules is 20, and the limit for egress rules is 20. This limit includes both IPv4 and IPv6 rules, and includes the default deny rules (rule number 32767 for IPv4 and 32768 for IPv6, or an asterisk * in the Amazon VPC console). This limit can be increased upon request up to a maximum if 40; however, network performance may be impacted due to the increased workload to process the additional rules.
Active VPC peering connections per VPC	50	To increase this limit, contact AWS Support. The maximum limit is 125 peering connections per VPC. The number of entries per route table should be increased accordingly; however, network performance may be impacted.
Outstanding VPC peering connection requests	25	This is the limit for the number of outstanding VPC peering connection requests that you've requested from your account. To increase this limit, contact AWS Support.

Resource	Default limit	Comments
Expiry time for an unaccepted VPC peering connection request	1 week (168 hours)	To increase this limit, contact AWS Support.
VPC endpoints per region	20	To increase this limit, contact AWS Support. The maximum limit is 255 endpoints per VPC, regardless of your endpoint limit per region.
Flow logs per single network interface, single subnet, or single VPC in a region	2	You can effectively have 6 flow logs per network interface if you create 2 flow logs for the subnet, and 2 flow logs for the VPC in which your network interface resides. This limit cannot be increased.
NAT gateways per Availability Zone	5	To increase this limit, submit a request. A NAT gateway in the `pending`, `active`, or `deleting` state counts against your limit.

For information about additional documented limits, see Amazon VPC Limits in the *Amazon VPC User Guide*.

AWS WAF Limits

AWS WAF has default limits on the number of entities per account. You can request an increase in these limits.

Resource	Default Limit
Web ACLs per AWS account	10
Rules per AWS account	50
Conditions per AWS account	50
Requests per Second	10,000 per web ACL*
IP address ranges (in CIDR notation) per IP match condition	1000

*This limit applies only to AWS WAF on an Application Load Balancer. Requests per Second (RPS) limits for AWS WAF on CloudFront are the same as the RPS limits support by CloudFront described in the CloudFront developer guide.

The following limits on AWS WAF entities can't be changed.

Resource	Limit
Rules per web ACL	10
Conditions per rule	10
Filters per cross-site scripting match condition	10

Resource	Limit
Filters per size constraint condition	10
Filters per SQL injection match condition	10
Filters per string match condition	10
In string match conditions, the number of characters in HTTP header names, when you've configured AWS WAF to inspect the headers in web requests for a specified value	40
In string match conditions, the number of bytes in the value tfor which AWS WAF should search	50

These limits are the same for all regions in which AWS WAF is available. Each region is subject to these limits individually. That is, the limits are not cumulative across regions.

Amazon WorkSpaces Limits

Resource	Default Limit
WorkSpaces	1
Images	5

AWS IP Address Ranges

Amazon Web Services (AWS) publishes its current IP address ranges in JSON format. To view the current ranges, download the `.json` file. To maintain history, save successive versions of the `.json` file on your system. To determine whether there have been changes since the last time that you saved the file, check the publication time in the current file and compare it to the publication time in the last file that you saved.

Contents

Download

Download ip-ranges.json

If you access this file programmatically, it is your responsibility to ensure that the application downloads the file only after successfully verifying the TLS certificate presented by the server.

Syntax

The syntax of `ip-ranges.json` is as follows.

```
{
  "syncToken": "0123456789",
  "createDate": "yyyy-mm-dd-hh-mm-ss",
  "prefixes": [
    {
      "ip_prefix": "cidr",
      "region": "region",
      "service": "subset"
    }
  ],
  "ipv6_prefixes": [
    {
      "ipv6_prefix": "cidr",
      "region": "region",
      "service": "subset"
    }
  ]
}
```

syncToken

The publication time, in Unix epoch time format.
Type: String

Example: `"syncToken": "1416435608"`

createDate

The publication date and time.
Type: String

Example: `"createDate": "2014-11-19-23-29-02"`

prefixes

The IP prefixes for the IPv4 address ranges.
Type: Array

ipv6_prefixes

The IP prefixes for the IPv6 address ranges.
Type: Array

ip_prefix

The public IPv4 address range, in CIDR notation. Note that AWS may advertise a prefix in more specific ranges. For example, prefix 96.127.0.0/17 in the file may be advertised as 96.127.0.0/21, 96.127.8.0/21, 96.127.32.0/19, and 96.127.64.0/18.
Type: String

Example: `"ip_prefix": "198.51.100.2/24"`

ipv6_prefix

The public IPv6 address range, in CIDR notation. Note that AWS may advertise a prefix in more specific ranges.
Type: String

Example: `"ipv6_prefix": "2001:db8:1234::/64"`

region

The AWS region or `GLOBAL` for edge locations. Note that the `CLOUDFRONT` and `ROUTE53` ranges are `GLOBAL`. You should ignore any values other than the values listed here.
Type: String

Valid values: `ap-northeast-1` | `ap-northeast-2` | `ap-south-1` | `ap-southeast-1` | `ap-southeast-2` | `cn-north-1` | `eu-central-1` | `eu-west-1` | `sa-east-1` | `us-east-1` | `us-gov-west-1` | `us-west-1` | `us-west-2` | `GLOBAL`

Example: `"region": "us-east-1"`

service

The subset of IP address ranges. Specify `AMAZON` to get all IP address ranges (for example, the ranges in the `EC2` subset are also in the `AMAZON` subset). Note that some IP address ranges are only in the `AMAZON` subset. You should ignore any values other than the values listed here.
Type: String

Valid values: `AMAZON` | `EC2` | `CLOUDFRONT` | `ROUTE53` | `ROUTE53_HEALTHCHECKS` | `S3`

Example: `"service": "AMAZON"`

Filtering the JSON File

You can download a command line tool to help you filter the information to just what you are looking for.

Windows

The AWS Tools for Windows PowerShell includes a cmdlet, `Get-AWSPublicIpAddressRange`, to parse this JSON file. The following examples demonstrate its use. For more information, see Querying the Public IP Address Ranges for AWS.

Example 1. Get the creation date

```
PS C:\> Get-AWSPublicIpAddressRange -OutputPublicationDate

Thursday, February 18, 2016 5:22:15 PM
```

Example 2. Get the information for a specific region

```
PS C:\> Get-AWSPublicIpAddressRange -Region us-east-1

IpPrefix        Region       Service
--------        ------       -------
23.20.0.0/14    us-east-1    AMAZON
50.16.0.0/15    us-east-1    AMAZON
50.19.0.0/16    us-east-1    AMAZON
...
```

Example 3. Get all IP addresses

```
PS C:\> (Get-AWSPublicIpAddressRange).IpPrefix

23.20.0.0/14
27.0.0.0/22
43.250.192.0/24
...
```

Linux

The following example commands use the jq tool to parse a local copy of the JSON file.

Example 1. Get the creation date

```
$ jq .createDate < ipranges.json

"2016-02-18-17-22-15"
```

Example 2. Get the information for a specific region

```
$ jq  '.prefixes[] | select(.region=="us-east-1")' < ipranges.json

{
  "ip_prefix": "23.20.0.0/14",
  "region": "us-east-1",
  "service": "AMAZON"
},
{
  "ip_prefix": "50.16.0.0/15",
  "region": "us-east-1",
  "service": "AMAZON"
},
{
  "ip_prefix": "50.19.0.0/16",
  "region": "us-east-1",
  "service": "AMAZON"
},
...
```

Example 3. Get all IP addresses

```
$ jq -r '.prefixes | .[].ip_prefix' < ipranges.json

23.20.0.0/14
27.0.0.0/22
43.250.192.0/24
...
```

Implementing Egress Control

To allow an instance to access only AWS services, create a security group with rules that allow outbound traffic to the CIDR blocks in the `AMAZON` list, minus the CIDR blocks that are also in the `EC2` list.

AWS IP Address Ranges Notifications

Whenever there is a change to the AWS IP address ranges, we send notifications to subscribers of the `AmazonIpSpaceChanged` topic. The payload contains information in the following format:

```
{
  "create-time":"yyyy-mm-ddThh:mm:ss+00:00",
  "synctoken":"0123456789",
  "md5":"6a45316e8bc9463c9e926d5d37836d33",
  "url":"https://ip-ranges.amazonaws.com/ip-ranges.json"
}
```

create-time

The creation date and time.

Notifications could be delivered out of order. Therefore, we recommend that you check the timestamps to ensure the correct order.

synctoken

The publication time, in Unix epoch time format.

md5

The cryptographic hash value of the `ip-ranges.json` file. You can use this value to check whether the downloaded file is corrupted.

url

The location of the `ip-ranges.json` file.

If you want to be notified whenever there is a change to the AWS IP address ranges, you can subscribe as follows to receive notifications using Amazon SNS.

To subscribe to AWS IP address range notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v2/home.
2. In the navigation bar, change the region to **US East (N. Virginia)**, if necessary. You must select this region because the SNS notifications that you are subscribing to were created in this region.
3. In the navigation pane, choose **Subscriptions**.
4. Choose **Create subscription**.
5. In the **Create subscription** dialog box, do the following:

    a. For **Topic ARN**, copy the following Amazon Resource Name (ARN):

    ```
    arn:aws:sns:us-east-1:806199016981:AmazonIpSpaceChanged
    ```

    b. For **Protocol**, choose the protocol to use (for example, `Email`).

    c. For **Endpoint**, type the endpoint to receive the notification (for example, your email address).

    d. Choose **Create subscription**.
6. You'll be contacted on the endpoint that you specified and asked to confirm your subscription. For example, if you specified an email address, you'll receive an email message with the subject line `AWS Notification - Subscription Confirmation`. Follow the directions to confirm your subscription.

Notifications are subject to the availability of the endpoint. Therefore, you might want to check the JSON file periodically to ensure that you've got the latest ranges. For more information about Amazon SNS reliability, see https://aws.amazon.com/sns/faqs/#Reliability.

If you no longer want to receive these notifications, use the following procedure to unsubscribe.

To unsubscribe from AWS IP address ranges notifications

1. Open the Amazon SNS console at https://console.aws.amazon.com/sns/v2/home.
2. In the navigation pane, choose **Subscriptions**.
3. Select the check box for the subscription.
4. Choose **Actions**, **Delete subscriptions**.
5. When prompted for confirmation, choose **Delete**.

For more information about Amazon SNS, see the *Amazon Simple Notification Service Developer Guide*.

Error Retries and Exponential Backoff in AWS

Numerous components on a network, such as DNS servers, switches, load balancers, and others can generate errors anywhere in the life of a given request. The usual technique for dealing with these error responses in a networked environment is to implement retries in the client application. This technique increases the reliability of the application and reduces operational costs for the developer.

Each AWS SDK implements automatic retry logic. The AWS SDK for Java automatically retries requests, and you can configure the retry settings using the `ClientConfiguration` class. For example, you might want to turn off the retry logic for a web page that makes a request with minimal latency and no retries. Use the `ClientConfiguration` class and provide a `maxErrorRetry` value of `0` to turn off the retries.

If you're not using an AWS SDK, you should retry original requests that receive server (5xx) or throttling errors. However, client errors (4xx) indicate that you need to revise the request to correct the problem before trying again.

In addition to simple retries, each AWS SDK implements exponential backoff algorithm for better flow control. The idea behind exponential backoff is to use progressively longer waits between retries for consecutive error responses. You should implement a maximum delay interval, as well as a maximum number of retries. The maximum delay interval and maximum number of retries are not necessarily fixed values, and should be set based on the operation being performed, as well as other local factors, such as network latency.

Most exponential backoff algorithms use jitter (randomized delay) to prevent successive collisions. Because you aren't trying to avoid such collisions in these cases, you don't need to use this random number. However, if you use concurrent clients, jitter can help your requests succeed faster. For more information, see the blog post for Exponential Backoff and Jitter.

The following pseudo code shows one way to poll for a status using an incremental delay.

```
Do some asynchronous operation.

retries = 0

DO
    wait for (2^retries * 100) milliseconds

    status = Get the result of the asynchronous operation.
    IF status = SUCCESS
        retry = false
    ELSE IF status = NOT_READY
        retry = true
    ELSE IF status = THROTTLED
        retry = true
    ELSE
        Some other error occurred, so stop calling the API.
        retry = false
    END IF

    retries = retries + 1

WHILE (retry AND (retries < MAX_RETRIES))
```

The following code demonstrates how to implement this incremental delay in Java.

```
public enum Results {
    SUCCESS,
    NOT_READY,
    THROTTLED,
    SERVER_ERROR
}

/*
 * Performs an asynchronous operation, then polls for the result of the
 * operation using an incremental delay.
 */
public static void doOperationAndWaitForResult() {

    try {
        // Do some asynchronous operation.
        long token = asyncOperation();

        int retries = 0;
        boolean retry = false;

        do {
            long waitTime = Math.min(getWaitTimeExp(retries), MAX_WAIT_INTERVAL);

            System.out.print(waitTime + "\n");

            // Wait for the result.
            Thread.sleep(waitTime);

            // Get the result of the asynchronous operation.
            Results result = getAsyncOperationResult(token);

            if (Results.SUCCESS == result) {
                retry = false;
            } else if (Results.NOT_READY == result) {
                retry = true;
            } else if (Results.THROTTLED == result) {
                retry = true;
            } else if (Results.SERVER_ERROR == result) {
                retry = true;
            }
            else {
                // Some other error occurred, so stop calling the API.
                retry = false;
            }

        } while (retry && (retries++ < MAX_RETRIES));
    }
    catch (Exception ex) {
    }
}

/*
 * Returns the next wait interval, in milliseconds, using an exponential
 * backoff algorithm.
 */
public static long getWaitTimeExp(int retryCount) {

    long waitTime = ((long) Math.pow(2, retryCount) * 100L);

    return waitTime;
}
```

AWS Command Line Tools

AWS Command Line Interface (AWS CLI)

Amazon Web Services (AWS) offers the AWS Command Line Interface (AWS CLI), a single tool for controlling and managing multiple AWS services. To download the AWS CLI or to view the list of supported services, see AWS Command Line Interface.

AWS also offers the AWS Tools for Windows PowerShell for those who script in the PowerShell environment.

Previous AWS Command Line Interface Tools

The prior AWS CLI tools are still available. If you need the prior AWS CLI tools, see the following table, which provides links to the command line tools and their documentation.

Product	Download	Documentation
Auto Scaling	Download Page: Auto Scaling Command Line Tools	Auto Scaling Command Line Tools Quick Reference Card
AWS CloudFormation	Download Page: AWS CloudFormation Command Line Tools	AWS CloudFormation Command Line Tools Reference AWS CloudFormation Command Line Tools Quick Reference Card
Amazon CloudSearch	Download Page: Amazon CloudSearch Command Line Tools for Windows Download Page: Amazon CloudSearch Command Line Tools for Mac OS/Linux	Amazon CloudSearch Developer Guide

Product	Download	Documentation
AWS Elastic Beanstalk	Download Page: AWS Elastic Beanstalk Command Line Tools	AWS Elastic Beanstalk Command Line Tools Reference
Amazon Elastic Compute Cloud	Download Page: Amazon EC2 API Command Line Tools Download Page: Amazon EC2 AMI Command Line Tools	Amazon EC2 Command Line Tools Reference Amazon EC2 Command Line Tools Quick Reference Card
Elastic Load Balancing	Download Page: Elastic Load Balancing Command Line Tools	Elastic Load Balancing Command Line Tools Quick Reference Card
Amazon EMR	Download Page: Amazon EMR Command Line Tools	Amazon EMR Command Line Tools Quick Reference Card
Amazon ElastiCache	Download Page: Amazon ElastiCache Command Line Tools	Amazon ElastiCache Command Line Tools Reference
AWS Identity and Access Management	The IAM command line tools package is deprecated. To perform IAM actions at the command line, use the AWS Command Line Interface.	AWS CLI User Guide AWS Identity and Access Management from the AWS Command Line Interface IAM reference in the AWS CLI
AWS Import/Export Disk	Download Page: Download the AWS Import/Export Disk Web Service Tool	What Is AWS Import/Export Disk?
Amazon Redshift	Download Page: AWS Command Line Interface	Amazon Redshift reference in the AWS CLI

Product	Download	Documentation
Amazon Relational Database Service	Download Page: Amazon RDS Command Line Tools	Amazon RDS Command Line Tools Reference Amazon RDS Command Line Tools Quick Reference Card
Amazon Simple Email Service	Download Page: Amazon SES Command Line Tools	Amazon SES Command Line Tools Documentation
Amazon Simple Notification Service	Download Page: Amazon SNS Command Line Tools	Amazon SNS Command Line Tools Reference
Amazon Virtual Private Cloud	Download Page: Amazon EC2 Command Line Tools	Amazon EC2 Command Line Tools Reference Amazon VPC Command Line Tools Quick Reference Card

Document Conventions

This section lists the common typographical conventions for AWS technical publications.

Typographical Conventions

This section describes common typographical conventions.

Convention	Description/Example
	A visual reference to further discussion elsewhere
`java -version`	Inline code (including commands, constants, XML elements, logical values, operations, parameters, and regular expressions)

`# ls -l /var/www/html/index.html`
`-rw-rw-r--  1 root root 1872 Jun 21 09:33 /var/www/html/index.html`
`# date`
`Wed Jun 21 09:33:42 EDT 2006`
	Blocks of sample code

`(start | stride | edge)`	Mutually exclusive options separated by vertical bars

`[-n, -quiet]`
`<CustomerId>[ID]</CustomerId>`
	Optional parameters
	-or-
	XML replaceable text

Convention	Description/Example
Amazon Machine Image (AMI) *Amazon EC2 User Guide for Linux Instances*	Important words or phrases -or- Technical publications
`MyPassword`	Text that the user types
On the **File** menu, choose **Properties**.	Console pages, menus, sections, or fields
For more information, see Document Conventions.	Link to other content

`your-s3-bucket`
`% ec2-register <your-s3-bucket>/image.manifest`
<your-S3-bucket>
	Placeholder text for a required value

CTRL + ENTER	Key names and key sequences

Documentation History

This guide was last updated on March 13, 2017.

The following table describes the important changes since the last release of the *Amazon Web Services General Reference*.

Change	Description	Release Date
EU (Frankfurt) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the EU (Frankfurt) Region.	March 7, 2017
EU (London) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the EU (London) Region.	December 13, 2016
Canada (Central) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the Canada (Central) Region.	December 8, 2016
US East (Ohio) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the US East (Ohio) Region.	October 17, 2016
Asia Pacific (Mumbai) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the Asia Pacific (Mumbai) Region.	June 27, 2016
Asia Pacific (Seoul) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the Asia Pacific (Seoul) Region.	January 6, 2016
EU (Frankfurt) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the EU (Frankfurt) Region.	October 23, 2014
South America (São Paulo) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the South America (São Paulo) Region.	December 14, 2011
US West (N. California) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the US West (N. California) Region.	November 8, 2011

Change	Description	Release Date
AWS GovCloud (US) Region	The AWS Regions and Endpoints (p. 2) topic has been updated to include information for the AWS GovCloud (US) Region, designed to meet the unique regulatory requirements of the United States Government.	August 16, 2011
AWS Command Line Tools	The AWS Command Line Tools (p. 181) topic has been added to provide links to the command line tools and their documentation for AWS products.	July 26, 2011
First release	This is the first release of the Amazon Web Services General Reference.	March 2, 2011

AWS Glossary

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

Numbers and Symbols

100-continue

A method that enables a client to see if a server can accept a request before actually sending it. For large PUT requests, this method can save both time and bandwidth charges.

A

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

AAD

See additional authenticated data.

access control list (ACL)

A document that defines who can access a particular bucket (p. 200) or object. Each bucket (p. 200) and object in Amazon S3 (p. 193) has an ACL. The document defines what each type of user can do, such as write and read permissions.

access identifiers

See credentials.

access key

The combination of an access key ID (p. 188) (like `AKIAIOSFODNN7EXAMPLE`) and a secret access key (p. 228) (like `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). You use access keys to sign API requests that you make to AWS.

access key ID

A unique identifier that's associated with a secret access key (p. 228); the access key ID and secret access key are used together to sign programmatic AWS requests cryptographically.

access key rotation

A method to increase security by changing the AWS access key ID. This method enables you to retire an old key at your discretion.

access policy language

A language for writing documents (that is, *policies* (p. 221)) that specify who can access a particular AWS resource (p. 225) and under what conditions.

account

A formal relationship with AWS that is associated with (1) the owner email address and password, (2) the control of resource (p. 225)s created under its umbrella, and (3) payment for the AWS activity related to those resources. The AWS account has permission to do anything and everything with all the AWS account resources. This is in contrast to a user (p. 234), which is an entity contained within the account.

account activity

A web page showing your month-to-date AWS usage and costs. The account activity page is located at https://aws.amazon.com/account-activity/.

ACL

See access control list (ACL).

ACM

See AWS Certificate Manager (ACM).

action

An API function. Also called *operation* or *call*. The activity the principal (p. 222) has permission to perform. The action is B in the statement "A has permission to do B to C where D applies." For example, Jane sends a request to Amazon SQS (p. 193) with Action=ReceiveMessage.

Amazon CloudWatch (p. 190): The response initiated by the change in an alarm's state: for example, from OK to ALARM. The state change may be triggered by a metric reaching the alarm threshold, or by a SetAlarmState request. Each alarm can have one or more actions assigned to each state. Actions are performed once each time the alarm changes to a state that has an action assigned, such as an Amazon Simple Notification Service (p. 193) notification, an Auto Scaling (p. 195) policy (p. 221) execution or an Amazon EC2 (p. 191) instance (p. 213) stop/terminate action.

active trusted signers

A list showing each of the trusted signers you've specified and the IDs of the corresponding active key pairs that Amazon CloudFront (p. 190) is aware of. To be able to create working signed URLs, a trusted signer must appear in this list with at least one key pair ID.

additional authenticated data

Information that is checked for integrity but not encrypted, such as headers or other contextual metadata.

administrative suspension

Auto Scaling (p. 195) might suspend processes for Auto Scaling group (p. 195) that repeatedly fail to launch instances. Auto Scaling groups that most commonly experience administrative suspension have zero running instances, have been trying to launch instances for more than 24 hours, and have not succeeded in that time.

alarm

An item that watches a single metric over a specified time period, and triggers an Amazon SNS (p. 193) topic (p. 233) or an Auto Scaling (p. 195) policy (p. 221) if the value of the metric crosses a threshold value over a predetermined number of time periods.

allow

One of two possible outcomes (the other is deny (p. 206)) when an IAM (p. 197) access policy (p. 221) is evaluated. When a user makes a request to AWS, AWS evaluates the request based on all permissions that apply to the user and then returns either allow or deny.

Amazon API Gateway

A fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.
See Also https://aws.amazon.com/api-gateway.

Amazon AppStream

A web service for streaming existing Windows applications from the cloud to any device.
See Also https://aws.amazon.com/appstream/.

Amazon Aurora

A fully managed MySQL-compatible relational database engine that combines the speed and availability of commercial databases with the simplicity and cost-effectiveness of open source databases.
See Also https://aws.amazon.com/rds/aurora/.

Amazon CloudFront

An AWS content delivery service that helps you improve the performance, reliability, and availability of your websites and applications.
See Also https://aws.amazon.com/cloudfront.

Amazon CloudSearch

A fully managed service in the AWS cloud that makes it easy to set up, manage, and scale a search solution for your website or application.

Amazon CloudWatch

A web service that enables you to monitor and manage various metrics, and configure alarm actions based on data from those metrics.
See Also https://aws.amazon.com/cloudwatch.

Amazon CloudWatch Events

A web service that enables you to deliver a timely stream of system events that describe changes in AWS resource (p. 225)s to AWS Lambda (p. 197) functions, streams in Amazon Kinesis Streams (p. 192), Amazon Simple Notification Service (p. 193) topics, or built-in targets.
See Also https://aws.amazon.com/cloudwatch.

Amazon CloudWatch Logs

A web service for monitoring and troubleshooting your systems and applications from your existing system, application, and custom log files. You can send your existing log files to CloudWatch Logs and monitor these logs in near real-time.
See Also https://aws.amazon.com/cloudwatch.

Amazon Cognito

A web service that makes it easy to save mobile user data, such as app preferences or game state, in the AWS cloud without writing any back-end code or managing any infrastructure. Amazon Cognito offers mobile identity management and data synchronization across devices.
See Also https://aws.amazon.com/cognito/.

Amazon DevPay

An easy-to-use online billing and account management service that makes it easy for you to sell an Amazon EC2 (p. 191) AMI (p. 192) or an application built on Amazon S3 (p. 193).
See Also https://aws.amazon.com/devpay.

Amazon DynamoDB

A fully managed NoSQL database service that provides fast and predictable performance with seamless scalability.
See Also https://aws.amazon.com/dynamodb/.

Amazon DynamoDB Storage Backend for Titan

A storage backend for the Titan graph database implemented on top of Amazon DynamoDB. Titan is a scalable graph database optimized for storing and querying graphs.
See Also https://aws.amazon.com/dynamodb/.

Amazon DynamoDB Streams

An AWS service that captures a time-ordered sequence of item-level modifications in any Amazon DynamoDB table, and stores this information in a log for up to 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near real time.
See Also https://aws.amazon.com/dynamodb/.

Amazon Elastic Block Store (Amazon EBS)

A service that provides block level storage volume (p. 235)s for use with EC2 instance (p. 207)s.

See Also https://aws.amazon.com/ebs.

Amazon EBS-backed AMI

A type of Amazon Machine Image (AMI) (p. 192) whose instance (p. 213)s use an Amazon EBS (p. 190) volume (p. 235) as their root device. Compare this with instances launched from instance store-backed AMI (p. 213)s, which use the instance store (p. 213) as the root device.

Amazon EC2 Container Registry (Amazon ECR)

A fully managed Docker container registry that makes it easy for developers to store, manage, and deploy Docker container images. Amazon ECR is integrated with Amazon EC2 Container Service (Amazon ECS) (p. 191) and AWS Identity and Access Management (IAM) (p. 197).

See Also https://aws.amazon.com/ecr.

Amazon EC2 Container Service (Amazon ECS)

A highly scalable, fast, container (p. 203) management service that makes it easy to run, stop, and manage Docker containers on a cluster (p. 202) of EC2 instance (p. 207)s.

See Also https://aws.amazon.com/ecs.

Amazon ECS service

A service for running and maintaining a specified number of task (p. 233)s (instantiations of a task definition (p. 233)) simultaneously.

Amazon EC2 VM Import Connector

See https://aws.amazon.com/ec2/vm-import.

Amazon Elastic Compute Cloud (Amazon EC2)

A web service that enables you to launch and manage Linux/UNIX and Windows server instance (p. 213)s in Amazon's data centers.

See Also https://aws.amazon.com/ec2.

Amazon Elastic File System (Amazon EFS)

A file storage service for EC2 (p. 191) instance (p. 213)s. Amazon EFS is easy to use and provides a simple interface with which you can create and configure file systems. Amazon EFS storage capacity grows and shrinks automatically as you add and remove files.

See Also https://aws.amazon.com/efs/.

Amazon EMR (Amazon EMR)

A web service that makes it easy to process large amounts of data efficiently. Amazon EMR uses Hadoop (p. 211) processing combined with several AWS products to do such tasks as web indexing, data mining, log file analysis, machine learning, scientific simulation, and data warehousing.

See Also https://aws.amazon.com/elasticmapreduce.

Amazon Elastic Transcoder

A cloud-based media transcoding service. Elastic Transcoder is a highly scalable tool for converting (or *transcoding*) media files from their source format into versions that will play on devices like smartphones, tablets, and PCs.

See Also https://aws.amazon.com/elastictranscoder/.

Amazon ElastiCache

A web service that simplifies deploying, operating, and scaling an in-memory cache in the cloud. The service improves the performance of web applications by providing information retrieval from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

See Also https://aws.amazon.com/elasticache/.

Amazon Elasticsearch Service (Amazon ES)

An AWS-managed service for deploying, operating, and scaling Elasticsearch, an open-source search and analytics engine, in the AWS Cloud. Amazon Elasticsearch Service (Amazon ES) also offers security options, high availability, data durability, and direct access to the Elasticsearch APIs.

See Also https://aws.amazon.com/elasticsearch-service.

Amazon GameLift

A managed service for deploying, operating, and scaling session-based multiplayer games.

See Also https://aws.amazon.com/gamelift/.

Amazon Glacier

A secure, durable, and low-cost storage service for data archiving and long-term backup. You can reliably store large or small amounts of data for significantly less than on-premises solutions. Amazon Glacier is optimized for infrequently accessed data, where a retrieval time of several hours is suitable.
See Also https://aws.amazon.com/glacier/.

Amazon Inspector

An automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Amazon Inspector automatically assesses applications for vulnerabilities or deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed report with prioritized steps for remediation.
See Also https://aws.amazon.com/inspector.

Amazon Kinesis

A platform for streaming data on AWS. Amazon Kinesis offers services that simplify the loading and analysis of streaming data.
See Also https://aws.amazon.com/kinesis/.

Amazon Kinesis Firehose

A fully managed service for loading streaming data into AWS. Firehose can capture and automatically load streaming data into Amazon S3 (p. 193) and Amazon Redshift (p. 192), enabling near real-time analytics with existing business intelligence tools and dashboards. Firehose automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, and encrypt the data before loading it.
See Also https://aws.amazon.com/kinesis/firehose/.

Amazon Kinesis Streams

A web service for building custom applications that process or analyze streaming data for specialized needs. Amazon Kinesis Streams can continuously capture and store terabytes of data per hour from hundreds of thousands of sources.
See Also https://aws.amazon.com/kinesis/streams/.

Amazon Lumberyard

A cross-platform, 3D game engine for creating high-quality games. You can connect games to the compute and storage of the AWS cloud and engage fans on Twitch.
See Also https://aws.amazon.com/lumberyard/.

Amazon Machine Image (AMI)

An encrypted machine image stored in Amazon Elastic Block Store (Amazon EBS) (p. 190) or Amazon Simple Storage Service (p. 193). AMIs are like a template of a computer's root drive. They contain the operating system and can also include software and layers of your application, such as database servers, middleware, web servers, and so on.

Amazon Machine Learning

A cloud-based service that creates machine learning (ML) models by finding patterns in your data, and uses these models to process new data and generate predictions.
See Also http://aws.amazon.com/machine-learning/.

Amazon ML

See Amazon Machine Learning.

Amazon Mobile Analytics

A service for collecting, visualizing, understanding, and extracting mobile app usage data at scale.
See Also https://aws.amazon.com/mobileanalytics.

Amazon Redshift

A fully managed, petabyte-scale data warehouse service in the cloud. With Amazon Redshift you can analyze your data using your existing business intelligence tools.
See Also https://aws.amazon.com/redshift/.

Amazon Relational Database Service (Amazon RDS)

A web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.
See Also https://aws.amazon.com/rds.

Amazon Resource Name (ARN)

A standardized way to refer to an AWS resource (p. 225). For example: arn:aws:iam::123456789012:user/division_abc/subdivision_xyz/Bob.

Amazon Route 53

A web service you can use to create a new DNS service or to migrate your existing DNS service to the cloud.
See Also https://aws.amazon.com/route53.

Amazon S3

See Amazon Simple Storage Service (Amazon S3).

Amazon S3-Backed AMI

See instance store-backed AMI.

Amazon Silk

A next-generation web browser available only on Fire OS tablets and phones. Built on a split architecture that divides processing between the client and the AWS cloud, Amazon Silk is designed to create a faster, more responsive mobile browsing experience.

Amazon Simple Email Service (Amazon SES)

An easy-to-use, cost-effective email solution for applications.
See Also https://aws.amazon.com/ses.

Amazon Simple Notification Service (Amazon SNS)

A web service that enables applications, end-users, and devices to instantly send and receive notifications from the cloud.
See Also https://aws.amazon.com/sns.

Amazon Simple Queue Service (Amazon SQS)

Reliable and scalable hosted queues for storing messages as they travel between computers.
See Also https://aws.amazon.com/sqs.

Amazon Simple Storage Service (Amazon S3)

Storage for the Internet. You can use it to store and retrieve any amount of data at any time, from anywhere on the web.
See Also https://aws.amazon.com/s3.

Amazon Simple Workflow Service (Amazon SWF)

A fully managed service that helps developers build, run, and scale background jobs that have parallel or sequential steps. Amazon SWF is like a state tracker and task coordinator in the cloud.
See Also https://aws.amazon.com/swf/.

Amazon Virtual Private Cloud (Amazon VPC)

A web service for provisioning a logically isolated section of the AWS cloud where you can launch AWS resource (p. 225)s in a virtual network that you define. You control your virtual networking environment, including selection of your own IP address range, creation of subnet (p. 231)s, and configuration of route table (p. 226)s and network gateways.
See Also https://aws.amazon.com/vpc.

Amazon VPC

See Amazon Virtual Private Cloud (Amazon VPC).

Amazon Web Services (AWS)

An infrastructure web services platform in the cloud for companies of all sizes.
See Also https://aws.amazon.com/what-is-cloud-computing/.

Amazon WorkDocs

A managed, secure enterprise document storage and sharing service with administrative controls and feedback capabilities.
See Also https://aws.amazon.com/workdocs/.

Amazon WorkMail

A managed, secure business email and calendar service with support for existing desktop and mobile email clients.
See Also https://aws.amazon.com/workmail/.

Amazon WorkSpaces

A managed, secure desktop computing service for provisioning cloud-based desktops and providing users access to documents, applications, and resource (p. 225)s from supported devices.
See Also https://aws.amazon.com/workspaces/.

Amazon WorkSpaces Application Manager (Amazon WAM)

A web service for deploying and managing applications for Amazon WorkSpaces. Amazon WAM accelerates software deployment, upgrades, patching, and retirement by packaging Windows desktop applications into virtualized application containers.
See Also https://aws.amazon.com/workspaces/applicationmanager.

AMI

See Amazon Machine Image (AMI).

analysis scheme

Amazon CloudSearch (p. 190): Language-specific text analysis options that are applied to a text field to control stemming and configure stopwords and synonyms.

application

AWS Elastic Beanstalk (p. 196): A logical collection of components, including environments, versions, and environment configurations. An application is conceptually similar to a folder.

AWS CodeDeploy (p. 196): A name that uniquely identifies the application to be deployed. AWS CodeDeploy uses this name to ensure the correct combination of revision, deployment configuration, and deployment group are referenced during a deployment.

Application Billing

The location where your customers manage the Amazon DevPay products they've purchased. The web address is http://www.amazon.com/dp-applications.

application revision

AWS CodeDeploy (p. 196): An archive file containing source content—such as source code, web pages, executable files, and deployment scripts—along with an application specification file (p. 194). Revisions are stored in Amazon S3 (p. 193) bucket (p. 200)s or GitHub (p. 211) repositories. For Amazon S3, a revision is uniquely identified by its Amazon S3 object key and its ETag, version, or both. For GitHub, a revision is uniquely identified by its commit ID.

application specification file

AWS CodeDeploy (p. 196): A YAML-formatted file used to map the source files in an application revision to destinations on the instance; specify custom permissions for deployed files; and specify scripts to be run on each instance at various stages of the deployment process.

application version

AWS Elastic Beanstalk (p. 196): A specific, labeled iteration of an application that represents a functionally consistent set of deployable application code. A version points to an Amazon S3 (p. 193) object (a JAVA WAR file) that contains the application code.

AppSpec file

See application specification file.

AUC

Area Under a Curve. An industry-standard metric to evaluate the quality of a binary classification machine learning model. AUC measures the ability of the model to predict a higher score for positive examples, those that are “correct,” than for negative examples, those that are “incorrect.” The AUC metric returns a decimal value from 0 to 1. AUC values near 1 indicate an ML model that is highly accurate.

ARN

See Amazon Resource Name (ARN).

artifact

AWS CodePipeline (p. 196): A copy of the files or changes that will be worked upon by the pipeline.

asymmetric encryption

Encryption (p. 208) that uses both a public key and a private key.

asynchronous bounce

A type of bounce (p. 200) that occurs when a receiver (p. 224) initially accepts an email message for delivery and then subsequently fails to deliver it.

atomic counter

DynamoDB: A method of incrementing or decrementing the value of an existing attribute without interfering with other write requests.

attribute

A fundamental data element, something that does not need to be broken down any further. In DynamoDB, attributes are similar in many ways to fields or columns in other database systems.

Amazon Machine Learning: A unique, named property within an observation in a data set. In tabular data, such as spreadsheets or comma-separated values (.csv) files, the column headings represent the attributes, and the rows contain values for each attribute.

Aurora

See Amazon Aurora.

authenticated encryption

Encryption (p. 208) that provides confidentiality, data integrity, and authenticity assurances of the encrypted data.

authentication

The process of proving your identity to a system.

Auto Scaling

A web service designed to launch or terminate instance (p. 213)s automatically based on user-defined policies (p. 221), schedules, and health check (p. 211)s.
See Also https://aws.amazon.com//autoscaling.

Auto Scaling group

A representation of multiple EC2 instance (p. 207)s that share similar characteristics, and that are treated as a logical grouping for the purposes of instance scaling and management.

Availability Zone

A distinct location within a region (p. 224) that is insulated from failures in other Availability Zones, and provides inexpensive, low-latency network connectivity to other Availability Zones in the same region.

AWS

See Amazon Web Services (AWS).

AWS Application Discovery Service

A web service that helps you plan to migrate to AWS by identifying IT assets in a data center—including servers, virtual machines, applications, application dependencies, and network infrastructure.
See Also https://aws.amazon.com/about-aws/whats-new/2016/04/aws-application-discovery-service/.

AWS Billing and Cost Management

The AWS cloud computing model in which you pay for services on demand and use as much or as little at any given time as you need. While resource (p. 225)s are active under your account, you pay for the cost of allocating those resources and for any incidental usage associated with those resources, such as data transfer or allocated storage.
See Also https://aws.amazon.com/billing/new-user-faqs/.

AWS Certificate Manager (ACM)

A web service for provisioning, managing, and deploying Secure Sockets Layer/Transport Layer Security (p. 234) (SSL/TLS) certificates for use with AWS services.
See Also https://aws.amazon.com/certificate-manager/.

AWS CloudFormation

A service for writing or changing templates that create and delete related AWS resource (p. 225)s together as a unit.
See Also https://aws.amazon.com/cloudformation.

AWS CloudHSM

A web service that helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated hardware security module (HSM) appliances within the AWS cloud.
See Also https://aws.amazon.com/cloudhsm/.

AWS CloudTrail

A web service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the API caller, the time of the

API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.
See Also https://aws.amazon.com/cloudtrail/.

AWS CodeCommit

A fully managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories.
See Also https://aws.amazon.com/codecommit.

AWS CodeDeploy

A service that automates code deployments to any instance, including EC2 instance (p. 207)s and instance (p. 213)s running on-premises.
See Also https://aws.amazon.com/codedeploy.

AWS CodeDeploy agent

A software package that, when installed and configured on an instance, enables that instance to be used in AWS CodeDeploy deployments.

AWS CodePipeline

A continuous delivery service for fast and reliable application updates.
See Also https://aws.amazon.com/codepipeline.

AWS Command Line Interface (AWS CLI)

A unified downloadable and configurable tool for managing AWS services. Control multiple AWS services from the command line and automate them through scripts.
See Also https://aws.amazon.com/cli/.

AWS Config

A fully managed service that provides an AWS resource (p. 225) inventory, configuration history, and configuration change notifications for better security and governance. You can create rules that automatically check the configuration of AWS resources that AWS Config records.
See Also https://aws.amazon.com/config/.

AWS Database Migration Service

A web service that can help you migrate data to and from many widely used commercial and open-source databases.
See Also https://aws.amazon.com/dms.

AWS Data Pipeline

A web service for processing and moving data between different AWS compute and storage services, as well as on-premises data sources, at specified intervals.
See Also https://aws.amazon.com/datapipeline.

AWS Device Farm

An app testing service that allows developers to test Android, iOS, and Fire OS devices on real, physical phones and tablets that are hosted by AWS.
See Also https://aws.amazon.com/device-farm.

AWS Direct Connect

A web service that simplifies establishing a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your data center, office, or colocation environment.
See Also https://aws.amazon.com/directconnect.

AWS Directory Service

A managed service for connecting your AWS resource (p. 225)s to an existing on-premises Microsoft Active Directory or to set up and operate a new, standalone directory in the AWS cloud.
See Also https://aws.amazon.com/directoryservice.

AWS Elastic Beanstalk

A web service for deploying and managing applications in the AWS cloud without worrying about the infrastructure that runs those applications.
See Also https://aws.amazon.com/elasticbeanstalk.

AWS GovCloud (US)

An isolated AWS Region designed to host sensitive workloads in the cloud, ensuring that this work meets the US government's regulatory and compliance requirements. The AWS GovCloud (US) Region adheres to United States International Traffic in Arms Regulations (ITAR), Federal Risk and Authorization Management Program (FedRAMP) requirements, Department of Defense (DOD)

Cloud Security Requirements Guide (SRG) Levels 2 and 4, and Criminal Justice Information Services (CJIS) Security Policy requirements.
See Also https://aws.amazon.com/govcloud-us/.

AWS Identity and Access Management (IAM)

A web service that enables Amazon Web Services (AWS) (p. 193) customers to manage users and user permissions within AWS.
See Also https://aws.amazon.com/iam.

AWS Import/Export

A service for transferring large amounts of data between AWS and portable storage devices.
See Also https://aws.amazon.com/importexport.

AWS IoT

A managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices.
See Also https://aws.amazon.com/iot.

AWS Key Management Service (AWS KMS)

A managed service that simplifies the creation and control of encryption (p. 208) keys that are used to encrypt data.
See Also https://aws.amazon.com/kms.

AWS Lambda

A web service that lets you run code without provisioning or managing servers. You can run code for virtually any type of application or back-end service with zero administration. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.
See Also https://aws.amazon.com/lambda/.

AWS managed key

One of two types of customer master key (CMK) (p. 204)s in AWS Key Management Service (AWS KMS) (p. 197).

AWS managed policy

An IAM (p. 197) managed policy (p. 216) that is created and managed by AWS.

AWS Management Console

A graphical interface to manage compute, storage, and other cloud resource (p. 225)s.
See Also https://aws.amazon.com/console.

AWS Management Portal for vCenter

A web service for managing your AWS resource (p. 225)s using VMware vCenter. You install the portal as a vCenter plug-in within your existing vCenter environment. Once installed, you can migrate VMware VMs to Amazon EC2 (p. 191) and manage AWS resources from within vCenter.
See Also https://aws.amazon.com/ec2/vcenter-portal/.

AWS Marketplace

A web portal where qualified partners to market and sell their software to AWS customers. AWS Marketplace is an online software store that helps customers find, buy, and immediately start using the software and services that run on AWS.
See Also https://aws.amazon.com/partners/aws-marketplace/.

AWS Mobile Hub

An integrated console that for building, testing, and monitoring mobile apps.
See Also https://aws.amazon.com/mobile.

AWS Mobile SDK

A software development kit whose libraries, code samples, and documentation help you build high quality mobile apps for the iOS, Android, Fire OS, Unity, and Xamarin platforms.
See Also https://aws.amazon.com/mobile/sdk.

AWS OpsWorks

A configuration management service that helps you use Chef to configure and operate groups of instances and applications. You can define the application's architecture and the specification of each component including package installation, software configuration, and resource (p. 225)s such as storage. You can automate tasks based on time, load, lifecycle events, and more.

See Also https://aws.amazon.com/opsworks/.

AWS Organizations

An account management service that enables you to consolidate multiple AWS accounts into an organization that you create and centrally manage.
See Also https://aws.amazon.com/organizations/.

AWS SDK for Go

A software development kit for integrating your Go application with the full suite of AWS services.
See Also https://aws.amazon.com/sdk-for-go/.

AWS SDK for Java

A software development kit that provides Java APIs for many AWS services including Amazon S3 (p. 193), Amazon EC2 (p. 191), Amazon DynamoDB (p. 190), and more. The single, downloadable package includes the AWS Java library, code samples, and documentation.
See Also https://aws.amazon.com/sdkforjava/.

AWS SDK for JavaScript in the Browser

A software development kit for accessing AWS services from JavaScript code running in the browser. Authenticate users through Facebook, Google, or Login with Amazon using web identity federation. Store application data in Amazon DynamoDB (p. 190), and save user files to Amazon S3 (p. 193).
See Also https://aws.amazon.com/sdk-for-browser/.

AWS SDK for JavaScript in Node.js

A software development kit for accessing AWS services from JavaScript in Node.js. The SDK provides JavaScript objects for AWS services, including Amazon S3 (p. 193), Amazon EC2 (p. 191), Amazon DynamoDB (p. 190), and Amazon Simple Workflow Service (Amazon SWF) (p. 193) . The single, downloadable package includes the AWS JavaScript library and documentation.
See Also https://aws.amazon.com/sdk-for-node-js/.

AWS SDK for .NET

A software development kit that provides .NET API actions for AWS services including Amazon S3 (p. 193), Amazon EC2 (p. 191), IAM (p. 197), and more. You can download the SDK as multiple service-specific packages on NuGet.
See Also https://aws.amazon.com/sdkfornet/.

AWS SDK for PHP

A software development kit and open-source PHP library for integrating your PHP application with AWS services like Amazon S3 (p. 193), Amazon Glacier (p. 192), and Amazon DynamoDB (p. 190).
See Also https://aws.amazon.com/sdkforphp/.

AWS SDK for Python (Boto)

A software development kit for using Python to access AWS services like Amazon EC2 (p. 191), Amazon EMR (p. 191), Auto Scaling (p. 195), Amazon Kinesis (p. 192), AWS Lambda (p. 197), and more.
See Also http://boto.readthedocs.org/en/latest/.

AWS SDK for Ruby

A software development kit for accessing AWS services from Ruby. The SDK provides Ruby classes for many AWS services including Amazon S3 (p. 193), Amazon EC2 (p. 191), Amazon DynamoDB (p. 190). and more. The single, downloadable package includes the AWS Ruby Library and documentation.
See Also https://aws.amazon.com/sdkforruby/.

AWS Security Token Service (AWS STS)

A web service for requesting temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) (p. 197) users or for users that you authenticate (federated users (p. 210)).
See Also https://aws.amazon.com/iam/.

AWS Service Catalog

A web service that helps organizations create and manage catalogs of IT services that are approved for use on AWS. These IT services can include everything from virtual machine images, servers, software, and databases to complete multitier application architectures.
See Also https://aws.amazon.com/servicecatalog/.

AWS Storage Gateway

A web service that connects an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and AWS's storage infrastructure.
See Also https://aws.amazon.com/storagegateway/.

AWS Toolkit for Eclipse

An open-source plug-in for the Eclipse Java IDE that makes it easier for developers to develop, debug, and deploy Java applications using Amazon Web Services.
See Also https://aws.amazon.com/eclipse/.

AWS Toolkit for Visual Studio

An extension for Microsoft Visual Studio that helps developers develop, debug, and deploy .NET applications using Amazon Web Services.
See Also https://aws.amazon.com/visualstudio/.

AWS Tools for Windows PowerShell

A set of PowerShell cmdlets to help developers and administrators manage their AWS services from the Windows PowerShell scripting environment.
See Also https://aws.amazon.com/powershell/.

AWS Trusted Advisor

A web service that inspects your AWS environment and makes recommendations for saving money, improving system availability and performance, and helping to close security gaps.
See Also https://aws.amazon.com/premiumsupport/trustedadvisor/.

AWS VPN CloudHub

Enables secure communication between branch offices using a simple hub-and-spoke model, with or without a VPC (p. 235).

AWS WAF

A web application firewall service that controls access to content by allowing or blocking web requests based on criteria that you specify, such as header values or the IP addresses that the requests originate from. AWS WAF helps protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources.
See Also https://aws.amazon.com/waf/.

B

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

basic monitoring

Monitoring of AWS-provided metrics derived at a 5-minute frequency.

batch

See document batch.

BGP ASN

Border Gateway Protocol Autonomous System Number. A unique identifier for a network, for use in BGP routing. Amazon EC2 (p. 191) supports all 2-byte ASN numbers in the range of 1 – 65335, with the exception of 7224, which is reserved.

batch prediction

Amazon Machine Learning: An operation that processes multiple input data observations at one time (asynchronously). Unlike real-time predictions, batch predictions are not available until all predictions have been processed.
See Also real-time predictions.

billing

See AWS Billing and Cost Management.

binary attribute

Amazon Machine Learning: An attribute for which one of two possible values is possible. Valid positive values are 1, y, yes, t, and true answers. Valid negative values are 0, n, no, f, and false. Amazon Machine Learning outputs 1 for positive values and 0 for negative values.
See Also attribute.

binary classification model

Amazon Machine Learning: A machine learning model that predicts the answer to questions where the answer can be expressed as a binary variable. For example, questions with answers of “1” or “0”, “yes” or “no”, “will click” or “will not click” are questions that have binary answers. The result for a binary classification model is always either a “1” (for a “true” or affirmative answers) or a “0” (for a “false” or negative answers).

blacklist

A list of IP addresses, email addresses, or domains that an Internet service provider (p. 213) suspects to be the source of spam (p. 230). The ISP blocks incoming email from these addresses or domains.

block

A data set. Amazon EMR (p. 191) breaks large amounts of data into subsets. Each subset is called a data block. Amazon EMR assigns an ID to each block and uses a hash table to keep track of block processing.

block device

A storage device that supports reading and (optionally) writing data in fixed-size blocks, sectors, or clusters.

block device mapping

A mapping structure for every AMI (p. 192) and instance (p. 213) that specifies the block devices attached to the instance.

blue/green deployment

AWS CodeDeploy: A deployment method in which the instances in a deployment group (the original environment) are replaced by a different set of instances (the replacement environment).

bootstrap action

A user-specified default or custom action that runs a script or an application on all nodes of a job flow before Hadoop (p. 211) starts.

Border Gateway Protocol Autonomous System Number

See BGP ASN.

bounce

A failed email delivery attempt.

breach

Auto Scaling (p. 195): The condition in which a user-set threshold (upper or lower boundary) is passed. If the duration of the breach is significant, as set by a breach duration parameter, it can possibly start a scaling activity (p. 227).

bucket

Amazon Simple Storage Service (Amazon S3) (p. 193): A container for stored objects. Every object is contained in a bucket. For example, if the object named `photos/puppy.jpg` is stored in the `johnsmith` bucket, then authorized users can access the object with the URL `http://johnsmith.s3.amazonaws.com/photos/puppy.jpg`.

bucket owner

The person or organization that owns a bucket (p. 200) in Amazon S3 (p. 193). Just as Amazon is the only owner of the domain name Amazon.com, only one person or organization can own a bucket.

bundling

A commonly used term for creating an Amazon Machine Image (AMI) (p. 192). It specifically refers to creating instance store-backed AMI (p. 213)s.

C

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

cache cluster

A logical cache distributed over multiple cache node (p. 201)s. A cache cluster can be set up with a specific number of cache nodes.

cache cluster identifier

Customer-supplied identifier for the cache cluster that must be unique for that customer in an AWS region (p. 224).

cache engine version

The version of the Memcached service that is running on the cache node.

cache node

A fixed-size chunk of secure, network-attached RAM. Each cache node runs an instance of the Memcached service, and has its own DNS name and port. Multiple types of cache nodes are supported, each with varying amounts of associated memory.

cache node type

An EC2 instance (p. 207) type used to run the cache node.

cache parameter group

A container for cache engine parameter values that can be applied to one or more cache clusters.

cache security group

A group maintained by ElastiCache that combines ingress authorizations to cache nodes for hosts belonging to Amazon EC2 (p. 191) security group (p. 228)s specified through the console or the API or command line tools.

canned access policy

A standard access control policy that you can apply to a bucket (p. 200) or object. Options include: private, public-read, public-read-write, and authenticated-read.

canonicalization

The process of converting data into a standard format that a service such as Amazon S3 (p. 193) can recognize.

capacity

The amount of available compute size at a given time. Each Auto Scaling group (p. 195) is defined with a minimum and maximum compute size. A scaling activity (p. 227) increases or decreases the capacity within the defined minimum and maximum values.

cartesian product processor

A processor that calculates a cartesian product. Also known as a *cartesian data processor*.

cartesian product

A mathematical operation that returns a product from multiple sets.

certificate

A credential that some AWS products use to authenticate AWS account (p. 189)s and users. Also known as an X.509 certificate (p. 236) . The certificate is paired with a private key.

chargeable resources

Features or services whose use incurs fees. Although some AWS products are free, others include charges. For example, in an AWS CloudFormation (p. 195) stack (p. 230), AWS resource (p. 225)s that have been created incur charges. The amount charged depends on the usage load. Use the Amazon Web Services Simple Monthly Calculator at http://calculator.s3.amazonaws.com/calc5.html to estimate your cost prior to creating instances, stacks, or other resources.

CIDR block

Classless Inter-Domain Routing. An Internet protocol address allocation and route aggregation methodology.
See Also Classless Inter-Domain Routing in Wikipedia.

ciphertext

Information that has been encrypted (p. 208), as opposed to plaintext (p. 221), which is information that has not.

ClassicLink

A feature for linking an EC2-Classic instance (p. 213) to a VPC (p. 235), allowing your EC2-Classic instance to communicate with VPC instances using private IP addresses.
See Also link to VPC, unlink from VPC.

classification

In machine learning, a type of problem that seeks to place (classify) a data sample into a single category or “class.” Often, classification problems are modeled to choose one category (class) out of two. These are binary classification problems.

Problems where more than two categories (classes) are available are called "multiclass classification" problems.
See Also binary classification model, multiclass classification model.

cloud service provider
A company that provides subscribers with access to Internet-hosted computing, storage, and software services.

CloudHub
See AWS VPN CloudHub.

CLI
See AWS Command Line Interface (AWS CLI).

cluster
A logical grouping of container instance (p. 203)s that you can place task (p. 233)s on.

Amazon Elasticsearch Service (Amazon ES) (p. 191): A logical grouping of one or more data nodes, optional dedicated master nodes, and storage required to run Amazon Elasticsearch Service (Amazon ES) and operate your Amazon ES domain.
See Also data node, dedicated master node, node.

cluster compute instance
A type of instance (p. 213) that provides a great amount of CPU power coupled with increased networking performance, making it well suited for High Performance Compute (HPC) applications and other demanding network-bound applications.

cluster placement group
A logical cluster compute instance (p. 202) grouping to provide lower latency and high-bandwidth connectivity between the instance (p. 213)s.

cluster status
Amazon Elasticsearch Service (Amazon ES) (p. 191): An indicator of the health of a cluster. A status can be green, yellow, or red. At the shard level, green means that all shards are allocated to nodes in a cluster, yellow means that the primary shard is allocated but the replica shards are not, and red means that the primary and replica shards of at least one index are not allocated. The shard status determines the index status, and the index status determines the cluster status.

CMK
See customer master key (CMK).

CNAME
Canonical Name Record. A type of resource record (p. 225) in the Domain Name System (DNS) that specifies that the domain name is an alias of another, canonical domain name. More simply, it is an entry in a DNS table that lets you alias one fully qualified domain name to another.

complaint
The event in which a recipient (p. 224) who does not want to receive an email message clicks "Mark as Spam" within the email client, and the Internet service provider (p. 213) sends a notification to Amazon SES (p. 193).

compound query
Amazon CloudSearch (p. 190): A search request that specifies multiple search criteria using the Amazon CloudSearch structured search syntax.

condition
IAM (p. 197): Any restriction or detail about a permission. The condition is *D* in the statement "A has permission to do B to C where D applies."

AWS WAF (p. 199): A set of attributes that AWS WAF searches for in web requests to AWS resource (p. 225)s such as Amazon CloudFront (p. 190) distributions. Conditions can include values such as the IP addresses that web requests originate from or values in request headers. Based on the specified conditions, you can configure AWS WAF to allow or block web requests to AWS resources.

conditional parameter
See mapping.

configuration API
Amazon CloudSearch (p. 190): The API call that you use to create, configure, and manage search domains.

configuration template

A series of key–value pairs that define parameters for various AWS products so that AWS Elastic Beanstalk (p. 196) can provision them for an environment.

consistency model

The method a service uses to achieve high availability. For example, it could involve replicating data across multiple servers in a data center.
See Also eventual consistency.

console

See AWS Management Console.

consolidated billing

A feature of the AWS Organizations service for consolidating payment for multiple AWS accounts. You create an organization that contains your AWS accounts, and you use the master account of your organization to pay for all member accounts. You can see a combined view of AWS costs that are incurred by all accounts in your organization, and you can get detailed cost reports for individual accounts.

container

A Linux container that was created from a Docker image as part of a task (p. 233).

container definition

Specifies which Docker image (p. 206) to use for a container (p. 203), how much CPU and memory the container is allocated, and more options. The container definition is included as part of a task definition (p. 233).

container instance

An EC2 instance (p. 207) that is running the Amazon EC2 Container Service (Amazon ECS) (p. 191) agent and has been registered into a cluster (p. 202). Amazon ECS task (p. 233)s are placed on active container instances.

container registry

Stores, manages, and deploys Docker image (p. 206)s.

continuous delivery

A software development practice in which code changes are automatically built, tested, and prepared for a release to production.
See Also https://aws.amazon.com/devops/continuous-delivery/.

continuous integration

A software development practice in which developers regularly merge code changes into a central repository, after which automated builds and tests are run.
See Also https://aws.amazon.com/devops/continuous-integration/.

cooldown period

Amount of time during which Auto Scaling (p. 195) does not allow the desired size of the Auto Scaling group (p. 195) to be changed by any other notification from an Amazon CloudWatch (p. 190) alarm (p. 189).

core node

An EC2 instance (p. 207) that runs Hadoop (p. 211) map and reduce tasks and stores data using the Hadoop Distributed File System (HDFS). Core nodes are managed by the master node (p. 216), which assigns Hadoop tasks to nodes and monitors their status. The EC2 instances you assign as core nodes are capacity that must be allotted for the entire job flow run. Because core nodes store data, you can't remove them from a job flow. However, you can add more core nodes to a running job flow.

Core nodes run both the DataNodes and TaskTracker Hadoop daemons.

corpus

Amazon CloudSearch (p. 190): A collection of data that you want to search.

credential helper

AWS CodeCommit (p. 196): A program that stores credentials for repositories and supplies them to Git when making connections to those repositories. The AWS CLI (p. 196) includes a credential helper that you can use with Git when connecting to AWS CodeCommit repositories.

credentials

Also called *access credentials* or *security credentials*. In authentication and authorization, a system uses credentials to identify who is making a call and whether to allow the requested access. In AWS, these credentials are typically the access key ID (p. 188) and the secret access key (p. 228).

cross-account access

The process of permitting limited, controlled use of resource (p. 225)s in one AWS account (p. 189) by a user in another AWS account. For example, in AWS CodeCommit (p. 196) and AWS CodeDeploy (p. 196) you can configure cross-account access so that a user in AWS account A can access an AWS CodeCommit repository created by account B. Or a pipeline in AWS CodePipeline (p. 196) created by account A can use AWS CodeDeploy resources created by account B. In IAM (p. 197) you use a role (p. 226) to delegate (p. 205) temporary access to a user (p. 234) in one account to resources in another.

cross-region replication

A client-side solution for maintaining identical copies of Amazon DynamoDB (p. 190) tables across different AWS region (p. 224)s, in near real time.

customer gateway

A router or software application on your side of a VPN tunnel that is managed by Amazon VPC (p. 193). The internal interfaces of the customer gateway are attached to one or more devices in your home network. The external interface is attached to the virtual private gateway (p. 235) across the VPN tunnel.

customer managed policy

An IAM (p. 197) managed policy (p. 216) that you create and manage in your AWS account (p. 189).

customer master key (CMK)

The fundamental resource (p. 225) that AWS Key Management Service (AWS KMS) (p. 197) manages. CMKs can be either customer-managed keys or AWS-managed keys. Use CMKs inside AWS KMS to encrypt (p. 208) or decrypt up to 4 kilobytes of data directly or to encrypt generated data keys, which are then used to encrypt or decrypt larger amounts of data outside of the service.

D

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

dashboard

See service health dashboard.

data consistency

A concept that describes when data is written or updated successfully and all copies of the data are updated in all AWS region (p. 224)s. However, it takes time for the data to propagate to all storage locations. To support varied application requirements, Amazon DynamoDB (p. 190) supports both eventually consistent and strongly consistent reads.
See Also eventual consistency, eventually consistent read, strongly consistent read.

data node

Amazon Elasticsearch Service (Amazon ES) (p. 191): An Elasticsearch instance that holds data and responds to data upload requests.
See Also dedicated master node, node.

data schema

See schema.

data source

The database, file, or repository that provides information required by an application or database. For example, in AWS OpsWorks (p. 197), valid data sources include an instance (p. 213) for a stack's MySQL layer or a stack's Amazon RDS (p. 192) service layer. In Amazon Redshift (p. 192), valid data sources include text files in an Amazon S3 (p. 193) bucket (p. 200), in an Amazon EMR (p. 191) cluster, or on a remote host that a cluster can access through an SSH connection.
See Also datasource.

database engine

The database software and version running on the DB instance (p. 205).

database name

The name of a database hosted in a DB instance (p. 205). A DB instance can host multiple databases, but databases hosted by the same DB instance must each have a unique name within that instance.

datasource

Amazon Machine Learning (p. 192): An object that contains metadata about the input data. Amazon ML reads the input data, computes descriptive statistics on its attributes, and stores the statistics—along with a schema and other information—as part of the datasource object. Amazon ML uses datasources to train and evaluate a machine learning model and generate batch predictions.
See Also data source.

DB compute class

Size of the database compute platform used to run the instance.

DB instance

An isolated database environment running in the cloud. A DB instance can contain multiple user-created databases.

DB instance identifier

User-supplied identifier for the DB instance. The identifier must be unique for that user in an AWS region (p. 224).

DB parameter group

A container for database engine parameter values that apply to one or more DB instance (p. 205)s.

DB security group

A method that controls access to the DB instance (p. 205). By default, network access is turned off to DB instances. After ingress is configured for a security group (p. 228), the same rules apply to all DB instances associated with that group.

DB snapshot

A user-initiated point backup of a DB instance (p. 205).

Dedicated Host

A physical server with EC2 instance (p. 207) capacity fully dedicated to a user.

Dedicated Instance

An instance (p. 213) that is physically isolated at the host hardware level and launched within a VPC (p. 235).

dedicated master node

Amazon Elasticsearch Service (Amazon ES) (p. 191): An Elasticsearch instance that performs cluster management tasks, but does not hold data or respond to data upload requests. Amazon Elasticsearch Service (Amazon ES) uses dedicated master nodes to increase cluster stability.
See Also data node, node.

Dedicated Reserved Instance

An option that you purchase to guarantee that sufficient capacity will be available to launch Dedicated Instance (p. 205)s into a VPC (p. 235).

delegation

Within a single AWS account (p. 189): Giving AWS user (p. 234)s access to resource (p. 225)s in your AWS account.

Between two AWS accounts: Setting up a trust between the account that owns the resource (the trusting account), and the account that contains the users that need to access the resource (the trusted account).
See Also trust policy.

delete marker

An object with a key and version ID, but without content. Amazon S3 (p. 193) inserts delete markers automatically into versioned bucket (p. 200)s when an object is deleted.

deliverability

The likelihood that an email message will arrive at its intended destination.

deliveries

The number of email messages, sent through Amazon SES (p. 193), that were accepted by an Internet service provider (p. 213) for delivery to recipient (p. 224)s over a period of time.

deny

The result of a policy (p. 221) statement that includes deny as the effect, so that a specific action or actions are expressly forbidden for a user, group, or role. Explicit deny take precedence over explicit allow (p. 189).

deployment configuration

AWS CodeDeploy (p. 196): A set of deployment rules and success and failure conditions used by the service during a deployment.

deployment group

AWS CodeDeploy (p. 196): A set of individually tagged instance (p. 213)s, EC2 instance (p. 207)s in Auto Scaling group (p. 195)s, or both.

detailed monitoring

Monitoring of AWS-provided metrics derived at a 1-minute frequency.

Description property

A property added to parameters, resource (p. 225)s, resource properties, mappings, and outputs to help you to document AWS CloudFormation (p. 195) template elements.

dimension

A name–value pair (for example, InstanceType=m1.small, or EngineName=mysql), that contains additional information to identify a metric.

discussion forums

A place where AWS users can post technical questions and feedback to help accelerate their development efforts and to engage with the AWS community. The discussion forums are located at https://aws.amazon.com/forums/.

distribution

A link between an origin server (such as an Amazon S3 (p. 193) bucket (p. 200)) and a domain name, which CloudFront (p. 190) automatically assigns. Through this link, CloudFront identifies the object you have stored in your origin server (p. 220).

DKIM

DomainKeys Identified Mail. A standard that email senders use to sign their messages. ISPs use those signatures to verify that messages are legitimate. For more information, see http://www.dkim.org.

DNS

See Domain Name System.

Docker image

A layered file system template that is the basis of a Docker container (p. 203). Docker images can comprise specific operating systems or applications.

document

Amazon CloudSearch (p. 190): An item that can be returned as a search result. Each document has a collection of fields that contain the data that can be searched or returned. The value of a field can be either a string or a number. Each document must have a unique ID and at least one field.

document batch

Amazon CloudSearch (p. 190): A collection of add and delete document operations. You use the document service API to submit batches to update the data in your search domain.

document service API

Amazon CloudSearch (p. 190): The API call that you use to submit document batches to update the data in a search domain.

document service endpoint

Amazon CloudSearch (p. 190): The URL that you connect to when sending document updates to an Amazon CloudSearch domain. Each search domain has a unique document service endpoint that remains the same for the life of the domain.

domain

Amazon Elasticsearch Service (Amazon ES) (p. 191): The hardware, software, and data exposed by Amazon Elasticsearch Service (Amazon ES) endpoints.
An Amazon ES domain is a service wrapper around an Elasticsearch cluster. An Amazon ES domain encapsulates the engine instances that process Amazon ES requests, the indexed data that you want to search, snapshots of the domain, access policies, and metadata.
See Also cluster, Elasticsearch.

Domain Name System

A service that routes Internet traffic to websites by translating friendly domain names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other.

Donation button

An HTML-coded button to provide an easy and secure way for US-based, IRS-certified 501(c)3 nonprofit organizations to solicit donations.

DynamoDB stream

An ordered flow of information about changes to items in anAmazon DynamoDB (p. 190) table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.
See Also Amazon DynamoDB Streams.

E

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

EBS

See Amazon Elastic Block Store (Amazon EBS).

EC2

See Amazon Elastic Compute Cloud (Amazon EC2).

EC2 compute unit

An AWS standard for compute CPU and memory. You can use this measure to evaluate the CPU capacity of different EC2 instance (p. 207) types.

EC2 instance

A compute instance (p. 213) in the Amazon EC2 (p. 191) service. Other AWS services use the term *EC2 instance* to distinguish these instances from other types of instances they support.

ECR

See Amazon EC2 Container Registry (Amazon ECR).

ECS

See Amazon EC2 Container Service (Amazon ECS).

edge location

A site that CloudFront (p. 190) uses to cache copies of your content for faster delivery to users at any location.

EFS

See Amazon Elastic File System (Amazon EFS).

Elastic

A company that provides open-source solutions—including Elasticsearch, Logstash, Kibana, and Beats—that are designed to take data from any source and search, analyze, and visualize it in real time.

Amazon Elasticsearch Service (Amazon ES) is an AWS-managed service for deploying, operating, and scaling Elasticsearch in the AWS Cloud.
See Also Amazon Elasticsearch Service (Amazon ES), Elasticsearch.

Elastic Block Store

See Amazon Elastic Block Store (Amazon EBS).

Elastic IP address

A fixed (static) IP address that you have allocated in Amazon EC2 (p. 191) or Amazon VPC (p. 193) and then attached to an instance (p. 213). Elastic IP addresses are associated with your account, not a specific instance. They are *elastic* because you can easily allocate, attach, detach, and free them as your needs change. Unlike traditional static IP addresses, Elastic IP addresses allow you to mask instance or Availability Zone (p. 195) failures by rapidly remapping your public IP addresses to another instance.

Elastic Load Balancing

A web service that improves an application's availability by distributing incoming traffic between two or more EC2 instance (p. 207)s.
See Also https://aws.amazon.com/elasticloadbalancing.

elastic network interface

An additional network interface that can be attached to an instance (p. 213). ENIs include a primary private IP address, one or more secondary private IP addresses, an elastic IP address (optional), a MAC address, membership in specified security group (p. 228)s, a description, and a source/destination check flag. You can create an ENI, attach it to an instance, detach it from an instance, and attach it to another instance.

Elasticsearch

An open source, real-time distributed search and analytics engine used for full-text search, structured search, and analytics. Elasticsearch was developed by the Elastic company.

Amazon Elasticsearch Service (Amazon ES) is an AWS-managed service for deploying, operating, and scaling Elasticsearch in the AWS Cloud.
See Also Amazon Elasticsearch Service (Amazon ES), Elastic.

EMR

See Amazon EMR (Amazon EMR).

encrypt

To use a mathematical algorithm to make data unintelligible to unauthorized user (p. 234)s while allowing authorized users a method (such as a key or password) to convert the altered data back to its original state.

encryption context

A set of key–value pairs that contains additional information associated with AWS Key Management Service (AWS KMS) (p. 197)–encrypted information.

endpoint

A URL that identifies a host and port as the entry point for a web service. Every web service request contains an endpoint. Most AWS products provide regional endpoints to enable faster connectivity.

Amazon ElastiCache (p. 191): The DNS name of a cache node (p. 201).

Amazon RDS (p. 192): The DNS name of a DB instance (p. 205).

AWS CloudFormation (p. 195): The DNS name or IP address of the server that receives an HTTP request.

endpoint port

Amazon ElastiCache (p. 191): The port number used by a cache node (p. 201).

Amazon RDS (p. 192): The port number used by a DB instance (p. 205).

envelope encryption

The use of a master key and a data key to algorithmically protect data. The master key is used to encrypt and decrypt the data key and the data key is used to encrypt and decrypt the data itself.

environment

AWS Elastic Beanstalk (p. 196): A specific running instance of an application (p. 194). The application has a CNAME and includes an application version and a customizable configuration (which is inherited from the default container type).

AWS CodeDeploy (p. 196): Instances in a deployment group in a blue/green deployment. At the start of a blue/green deployment, the deployment group is made up of instances in the original environment. At the end of the deployment, the deployment group is made up of instances in the replacement environment.

environment configuration

A collection of parameters and settings that define how an environment and its associated resources behave.

ephemeral store

See instance store.

epoch

The date from which time is measured. For most Unix environments, the epoch is January 1, 1970.

evaluation

Amazon Machine Learning: The process of measuring the predictive performance of a machine learning (ML) model.

Also a machine learning object that stores the details and result of an ML model evaluation.

evaluation datasource

The data that Amazon Machine Learning uses to evaluate the predictive accuracy of a machine learning model.

eventual consistency

The method through which AWS products achieve high availability, which involves replicating data across multiple servers in Amazon's data centers. When data is written or updated and `Success` is returned, all copies of the data are updated. However, it takes time for the data to propagate to all storage locations. The data will eventually be consistent, but an immediate read might not show the change. Consistency is usually reached within seconds.
See Also data consistency, eventually consistent read, strongly consistent read.

eventually consistent read

A read process that returns data from only one region and might not show the most recent write information. However, if you repeat your read request after a short time, the response should eventually return the latest data.
See Also data consistency, eventual consistency, strongly consistent read.

eviction

The deletion by CloudFront (p. 190) of an object from an edge location (p. 207) before its expiration time. If an object in an edge location isn't frequently requested, CloudFront might evict the object (remove the object before its expiration date) to make room for objects that are more popular.

exbibyte

A contraction of exa binary byte, an exbibyte is 2^60 or 1,152,921,504,606,846,976 bytes. An exabyte (EB) is 10^18 or 1,000,000,000,000,000,000 bytes. 1,024 EiB is a zebibyte (p. 236).

expiration

For CloudFront (p. 190) caching, the time when CloudFront stops responding to user requests with an object. If you don't use headers or CloudFront distribution (p. 206) settings to specify how long you want objects to stay in an edge location (p. 207), the objects expire after 24 hours. The next time a user requests an object that has expired, CloudFront forwards the request to the origin (p. 220).

explicit launch permission

An Amazon Machine Image (AMI) (p. 192) launch permission granted to a specific AWS account (p. 189).

exponential backoff

A strategy that incrementally increases the wait between retry attempts in order to reduce the load on the system and increase the likelihood that repeated requests will succeed. For example, client applications might wait up to 400 milliseconds before attempting the first retry, up to 1600 milliseconds before the second, up to 6400 milliseconds (6.4 seconds) before the third, and so on.

expression

Amazon CloudSearch (p. 190): A numeric expression that you can use to control how search hits are sorted. You can construct Amazon CloudSearch expressions using numeric fields, other rank expressions, a document's default relevance score, and standard numeric operators and functions. When you use the `sort` option to specify an expression in a search request, the expression is evaluated for each search hit and the hits are listed according to their expression values.

F

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) |

O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) |
W (p. 236) | X, Y, Z (p. 236)

facet

Amazon CloudSearch (p. 190): An index field that represents a category that you want to use to refine and filter search results.

facet enabled

Amazon CloudSearch (p. 190): An index field option that enables facet information to be calculated for the field.

FBL

See feedback loop.

feature transformation

Amazon Machine Learning: The machine learning process of constructing more predictive input representations or “features” from the raw input variables to optimize a machine learning model’s ability to learn and generalize. Also known as *data transformation* or *feature engineering*.

federated identity management

Allows individuals to sign in to different networks or services, using the same group or personal credentials to access data across all networks. With identity federation in AWS, external identities (federated users) are granted secure access to resource (p. 225)s in an AWS account (p. 189) without having to create IAM user (p. 234)s. These external identities can come from a corporate identity store (such as LDAP or Windows Active Directory) or from a third party (such as Login with Amazon, Facebook, or Google). AWS federation also supports SAML 2.0.

federated user

See federated identity management.

federation

See federated identity management.

feedback loop

The mechanism by which a mailbox provider (for example, an Internet service provider (p. 213)) forwards a recipient (p. 224)'s complaint (p. 202) back to the sender (p. 228).

field weight

The relative importance of a text field in a search index. Field weights control how much matches in particular text fields affect a document's relevance score.

filter

A criterion that you specify to limit the results when you list or describe your Amazon EC2 (p. 191) resource (p. 225)s.

filter query

A way to filter search results without affecting how the results are scored and sorted. Specified with the Amazon CloudSearch (p. 190) `fq` parameter.

FIM

See federated identity management.

Firehose

See Amazon Kinesis Firehose.

format version

See template format version.

forums

See discussion forums.

function

See intrinsic function.

fuzzy search

A simple search query that uses approximate string matching (fuzzy matching) to correct for typographical errors and misspellings.

G

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209)
| G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) |
O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) |
W (p. 236) | X, Y, Z (p. 236)

geospatial search

A search query that uses locations specified as a latitude and longitude to determine matches and sort the results.

gibibyte

A contraction of giga binary byte, a gibibyte is 2^30 or 1,073,741,824 bytes. A gigabyte (GB) is 10^9 or 1,000,000,000 bytes. 1,024 GiB is a tebibyte (p. 233).

GitHub

A web-based repository that uses Git for version control.

global secondary index

An index with a partition key and a sort key that can be different from those on the table. A global secondary index is considered global because queries on the index can span all of the data in a table, across all partitions.
See Also local secondary index.

grant

AWS Key Management Service (AWS KMS) (p. 197): A mechanism for giving AWS principal (p. 222)s long-term permissions to use customer master key (CMK) (p. 204)s.

grant token

A type of identifier that allows the permissions in a grant (p. 211) to take effect immediately.

ground truth

The observations used in the machine learning (ML) model training process that include the correct value for the target attribute. To train an ML model to predict house sales prices, the input observations would typically include prices of previous house sales in the area. The sale prices of these houses constitute the ground truth.

group

A collection of IAM (p. 197) user (p. 234)s. You can use IAM groups to simplify specifying and managing permissions for multiple users.

H

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

Hadoop

Software that enables distributed processing for big data by using clusters and simple programming models. For more information, see http://hadoop.apache.org.

hard bounce

A persistent email delivery failure such as "mailbox does not exist."

hardware VPN

A hardware-based IPsec VPN connection over the Internet.

health check

A system call to check on the health status of each instance in an Auto Scaling (p. 195) group.

high-quality email

Email that recipients find valuable and want to receive. Value means different things to different recipients and can come in the form of offers, order confirmations, receipts, newsletters, etc.

highlights

Amazon CloudSearch (p. 190): Excerpts returned with search results that show where the search terms appear within the text of the matching documents.

highlight enabled

Amazon CloudSearch (p. 190): An index field option that enables matches within the field to be highlighted.

hit

A document that matches the criteria specified in a search request. Also referred to as a *search result*.

HMAC

Hash-based Message Authentication Code. A specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret key. You can use it to verify both the data integrity and the authenticity of a message at the same time. AWS calculates the HMAC using a standard, cryptographic hash algorithm, such as SHA-256.

hosted zone

A collection of resource record (p. 225) sets that Amazon Route 53 (p. 193) hosts. Like a traditional DNS zone file, a hosted zone represents a collection of records that are managed together under a single domain name.

HVM virtualization

Hardware Virtual Machine virtualization. Allows the guest VM to run as though it is on a native hardware platform, except that it still uses paravirtual (PV) network and storage drivers for improved performance.
See Also PV virtualization.

I

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

IAM

See AWS Identity and Access Management (IAM).

IAM group

See group.

IAM policy simulator

See policy simulator.

IAM role

See role.

IAM user

See user.

Identity and Access Management

See AWS Identity and Access Management (IAM).

identity provider (IdP)

An IAM (p. 197) entity that holds metadata about external identity providers.

IdP

See identity provider (IdP) .

image

See Amazon Machine Image (AMI).

import/export station

A machine that uploads or downloads your data to or from Amazon S3 (p. 193).

import log

A report that contains details about how AWS Import/Export (p. 197) processed your data.

in-place deployment

AWS CodeDeploy: A deployment method in which instances in a deployment group are taken offline, updated with the latest application revision, and then placed back online as part of the same deployment group.

index

See search index.

index field

A name–value pair that is included in an Amazon CloudSearch (p. 190) domain's index. An index field can contain text or numeric data, dates, or a location.

indexing options

Configuration settings that define an Amazon CloudSearch (p. 190) domain's index fields, how document data is mapped to those index fields, and how the index fields can be used.

inline policy

An IAM (p. 197) policy (p. 221) that is embedded in a single IAM user (p. 234), group (p. 211), or role (p. 226).

input data

Amazon Machine Learning: The observations that you provide to Amazon Machine Learning to train and evaluate a machine learning model and generate predictions.

instance

A copy of an Amazon Machine Image (AMI) (p. 192) running as a virtual server in the AWS cloud.

instance family

A general instance type (p. 213) grouping using either storage or CPU capacity.

instance group

A Hadoop (p. 211) cluster contains one master instance group that contains one master node (p. 216), a core instance group containing one or more core node (p. 203) and an optional task node (p. 233) instance group, which can contain any number of task nodes.

instance profile

A container that passes IAM (p. 197) role (p. 226) information to an EC2 instance (p. 207) at launch.

instance store

Disk storage that is physically attached to the host computer for an EC2 instance (p. 207), and therefore has the same lifespan as the instance. When the instance is terminated, you lose any data in the instance store.

instance store-backed AMI

A type of Amazon Machine Image (AMI) (p. 192) whose instance (p. 213)s use an instance store (p. 213) volume (p. 235) as the root device. Compare this with instances launched from Amazon EBS (p. 190)-backed AMIs, which use an Amazon EBS volume as the root device.

instance type

A specification that defines the memory, CPU, storage capacity, and hourly cost for an instance (p. 213). Some instance types are designed for standard applications, whereas others are designed for CPU-intensive, memory-intensive applications, and so on.

Internet gateway

Connects a network to the Internet. You can route traffic for IP addresses outside your VPC (p. 235) to the Internet gateway.

Internet service provider

A company that provides subscribers with access to the Internet. Many ISPs are also mailbox provider (p. 216)s. Mailbox providers are sometimes referred to as ISPs, even if they only provide mailbox services.

intrinsic function

A special action in a AWS CloudFormation (p. 195) template that assigns values to properties not available until runtime. These functions follow the format *Fn::Attribute*, such as `Fn::GetAtt`. Arguments for intrinsic functions can be parameters, pseudo parameters, or the output of other intrinsic functions.

IP address

A numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). All EC2 instance (p. 207)s are assigned two IP addresses at launch, which are directly mapped to each other through network address translation (NAT (p. 218)): a private IP address (following RFC 1918) and a public IP address. Instances launched in a VPC (p. 193) are assigned only a private IP address. Instances launched in your default VPC are assigned both a private IP address and a public IP address.

IP match condition

AWS WAF (p. 199): An attribute that specifies the IP addresses or IP address ranges that web requests originate from. Based on the specified IP addresses, you can configure AWS WAF to allow or block web requests to AWS resource (p. 225)s such as Amazon CloudFront (p. 190) distributions.

ISP

See Internet service provider.

issuer

The person who writes a policy (p. 221) to grant permissions to a resource (p. 225). The issuer (by definition) is always the resource owner. AWS does not permit Amazon SQS (p. 193) users to create policies for resources they don't own. If John is the resource owner, AWS authenticates John's identity when he submits the policy he's written to grant permissions for that resource.

item

A group of attributes that is uniquely identifiable among all of the other items. Items in Amazon DynamoDB (p. 190) are similar in many ways to rows, records, or tuples in other database systems.

J

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

job flow

Amazon EMR (p. 191): One or more step (p. 231)s that specify all of the functions to be performed on the data.

job ID

A five-character, alphanumeric string that uniquely identifies an AWS Import/Export (p. 197) storage device in your shipment. AWS issues the job ID in response to a `CREATE JOB` email command.

job prefix

An optional string that you can add to the beginning of an AWS Import/Export (p. 197) log file name to prevent collisions with objects of the same name. See Also key prefix.

JSON

JavaScript Object Notation. A lightweight data interchange format. For information about JSON, see http://www.json.org/.

junk folder

The location where email messages that various filters determine to be of lesser value are collected so that they do not arrive in the recipient (p. 224)'s inbox but are still accessible to the recipient. This is also referred to as a spam (p. 230) or bulk folder.

K

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

key

A credential that identifies an AWS account (p. 189) or user (p. 234) to AWS (such as the AWS secret access key (p. 228)).

Amazon Simple Storage Service (Amazon S3) (p. 193), Amazon EMR (Amazon EMR) (p. 191): The unique identifier for an object in a bucket (p. 200). Every object in a bucket has exactly one key. Because a bucket and key together uniquely identify each object, you can think of Amazon S3 as a basic data map between the *bucket* + *key*, and the object itself. You can uniquely address every object in Amazon S3 through the combination of the web service endpoint, bucket name, and key, as in this example: `http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl`, where `doc` is the name of the bucket, and `2006-03-01/AmazonS3.wsdl` is the key.

AWS Import/Export (p. 197): The name of an object in Amazon S3. It is a sequence of Unicode characters whose UTF-8 encoding cannot exceed 1024 bytes. If a key, for example, logPrefix + import-log-JOBID, is longer than 1024 bytes, AWS Elastic Beanstalk (p. 196) returns an `InvalidManifestField` error.

IAM (p. 197): In a policy (p. 221), a specific characteristic that is the basis for restricting access (such as the current time, or the IP address of the requester).

Tagging resources: A general tag (p. 232) label that acts like a category for more specific tag values. For example, you might have EC2 instance (p. 207) with the tag key of *Owner* and the tag value of *Jan*. You can tag an AWS resource (p. 225) with up to 10 key–value pairs. Not all AWS resources can be tagged.

key pair

A set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key.

key prefix

A logical grouping of the objects in a bucket (p. 200). The prefix value is similar to a directory name that enables you to store similar data under the same directory in a bucket.

kibibyte

A contraction of kilo binary byte, a kibibyte is 2^10 or 1,024 bytes. A kilobyte (KB) is 10^3 or 1,000 bytes. 1,024 KiB is a mebibyte (p. 217).

KMS

See AWS Key Management Service (AWS KMS).

L

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

labeled data

In machine learning, data for which you already know the target or “correct” answer.

launch configuration

A set of descriptive parameters used to create new EC2 instance (p. 207)s in an Auto Scaling (p. 195) activity.

A template that an Auto Scaling group (p. 195) uses to launch new EC2 instances. The launch configuration contains information such as the Amazon Machine Image (AMI) (p. 192) ID, the instance type, key pairs, security group (p. 228)s, and block device mappings, among other configuration settings.

launch permission

An Amazon Machine Image (AMI) (p. 192) attribute that allows users to launch an AMI.

lifecycle

The lifecycle state of the EC2 instance (p. 207) contained in an Auto Scaling group (p. 195). EC2 instances progress through several states over their lifespan; these include *Pending*, *InService*, *Terminating* and *Terminated*.

lifecycle action

An action that can be paused by Auto Scaling, such as launching or terminating an EC2 instance.

lifecycle hook

Enables you to pause Auto Scaling after it launches or terminates an EC2 instance so that you can perform a custom action while the instance is not in service.

link to VPC

The process of linking (or attaching) an EC2-Classic instance (p. 213) to a ClassicLink-enabled VPC (p. 235).
See Also ClassicLink, unlink from VPC.

load balancer

A DNS name combined with a set of ports, which together provide a destination for all requests intended for your application. A load balancer can distribute traffic to multiple application instances across every Availability Zone (p. 195) within a region (p. 224). Load balancers can span multiple Availability Zones within an Amazon EC2 (p. 191) region, but they cannot span multiple regions.

local secondary index

An index that has the same partition key as the table, but a different sort key. A local secondary index is local in the sense that every partition of a local secondary index is scoped to a table partition that has the same partition key value.
See Also local secondary index.

logical name

A case-sensitive unique string within an AWS CloudFormation (p. 195) template that identifies a resource (p. 225), mapping (p. 216), parameter, or output. In an AWS CloudFormation template, each parameter, resource (p. 225), property, mapping, and output must be declared with a unique logical name. You use the logical name when dereferencing these items using the `Ref` function.

M

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

Mail Transfer Agent (MTA)

Software that transports email messages from one computer to another by using a client-server architecture.

mailbox provider

An organization that provides email mailbox hosting services. Mailbox providers are sometimes referred to as Internet service provider (p. 213)s, even if they only provide mailbox services.

mailbox simulator

A set of email addresses that you can use to test an Amazon SES (p. 193)-based email sending application without sending messages to actual recipients. Each email address represents a specific scenario (such as a bounce or complaint) and generates a typical response that is specific to the scenario.

main route table

The default route table (p. 226) that any new VPC (p. 235) subnet (p. 231) uses for routing. You can associate a subnet with a different route table of your choice. You can also change which route table is the main route table.

managed policy

A standalone IAM (p. 197) policy (p. 221) that you can attach to multiple user (p. 234)s, group (p. 211)s, and role (p. 226)s in your IAM account (p. 189). Managed policies can either be AWS managed policies (which are created and managed by AWS) or customer managed policies (which you create and manage in your AWS account).

manifest

When sending a *create job* request for an import or export operation, you describe your job in a text file called a manifest. The manifest file is a YAML-formatted file that specifies how to transfer data between your storage device and the AWS cloud.

manifest file

Amazon Machine Learning: The file used for describing batch predictions. The manifest file relates each input data file with its associated batch prediction results. It is stored in the Amazon S3 output location.

mapping

A way to add conditional parameter values to an AWS CloudFormation (p. 195) template. You specify mappings in the template's optional Mappings section and retrieve the desired value using the `FN::FindInMap` function.

marker

See pagination token.

master node

A process running on an Amazon Machine Image (AMI) (p. 192) that keeps track of the work its core and task nodes complete.

maximum price

The maximum price you will pay to launch one or more Spot Instance (p. 230)s. If your maximum price exceeds the current Spot price (p. 230) and your restrictions are met, Amazon EC2 (p. 191) launches instances on your behalf.

maximum send rate

The maximum number of email messages that you can send per second using Amazon SES (p. 193).

mebibyte

A contraction of mega binary byte, a mebibyte is 2^20 or 1,048,576 bytes. A megabyte (MB) is 10^6 or 1,000,000 bytes. 1,024 MiB is a gibibyte (p. 211).

member resources

See resource.

message ID

Amazon Simple Email Service (Amazon SES) (p. 193): A unique identifier that is assigned to every email message that is sent.

Amazon Simple Queue Service (Amazon SQS) (p. 193): The identifier returned when you send a message to a queue.

metadata

Information about other data or objects. In Amazon Simple Storage Service (Amazon S3) (p. 193) and Amazon EMR (Amazon EMR) (p. 191) metadata takes the form of name–value pairs that describe the object. These include default metadata such as the date last modified and standard HTTP metadata such as Content-Type. Users can also specify custom metadata at the time they store an object. In Amazon Elastic Compute Cloud (Amazon EC2) (p. 191) metadata includes data about an EC2 instance (p. 207) that the instance can retrieve to determine things about itself, such as the instance type, the IP address, and so on.

metric

An element of time-series data defined by a unique combination of exactly one namespace (p. 218), exactly one metric name, and between zero and ten dimensions. Metrics and the statistics derived from them are the basis of Amazon CloudWatch (p. 190).

metric name

The primary identifier of a metric, used in combination with a namespace (p. 218) and optional dimensions.

MFA

See multi-factor authentication (MFA).

micro instance

A type of EC2 instance (p. 207) that is more economical to use if you have occasional bursts of high CPU activity.

MIME

See Multipurpose Internet Mail Extensions (MIME).

ML model

In machine learning (ML), a mathematical model that generates predictions by finding patterns in data. Amazon Machine Learning supports three types of ML models: binary classification, multiclass classification, and regression. Also known as a *predictive model*.
See Also binary classification model, multiclass classification model, regression model.

MTA

See Mail Transfer Agent (MTA).

Multi-AZ deployment

A primary DB instance (p. 205) that has a synchronous standby replica in a different Availability Zone (p. 195). The primary DB instance is synchronously replicated across Availability Zones to the standby replica.

multiclass classification model

A machine learning model that predicts values that belong to a limited, pre-defined set of permissible values. For example, "Is this product a book, movie, or clothing?"

multi-factor authentication (MFA)

An optional AWS account (p. 189) security feature. Once you enable AWS MFA, you must provide a six-digit, single-use code in addition to your sign-in

credentials whenever you access secure AWS webpages or the AWS Management Console (p. 197). You get this single-use code from an authentication device that you keep in your physical possession.
See Also https://aws.amazon.com/mfa/.

multi-valued attribute

An attribute with more than one value.

multipart upload

A feature that allows you to upload a single object as a set of parts.

Multipurpose Internet Mail Extensions (MIME)

An Internet standard that extends the email protocol to include non-ASCII text and nontext elements like attachments.

Multitool

A cascading application that provides a simple command-line interface for managing large datasets.

N

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

namespace

An abstract container that provides context for the items (names, or technical terms, or words) it holds, and allows disambiguation of homonym items residing in different namespaces.

NAT

Network address translation. A strategy of mapping one or more IP addresses to another while data packets are in transit across a traffic routing device. This is commonly used to restrict Internet communication to private instances while allowing outgoing traffic.
See Also Network Address Translation and Protocol Translation, NAT gateway, NAT instance.

NAT gateway

A NAT (p. 218) device, managed by AWS, that performs network address translation in a private subnet (p. 231), to secure inbound Internet traffic. A NAT gateway uses both NAT and port address translation.
See Also NAT instance.

NAT instance

A NAT (p. 218) device, configured by a user, that performs network address translation in a VPC (p. 235) public subnet (p. 231) to secure inbound Internet traffic.
See Also NAT gateway.

network ACL

An optional layer of security that acts as a firewall for controlling traffic in and out of a subnet (p. 231). You can associate multiple subnets with a single network ACL (p. 188), but a subnet can be associated with only one network ACL at a time.

Network Address Translation and Protocol Translation

(NAT (p. 218)-PT) An Internet protocol standard defined in RFC 2766.
See Also NAT instance, NAT gateway.

n-gram processor

A processor that performs n-gram transformations.
See Also n-gram transformation.

n-gram transformation

Amazon Machine Learning: A transformation that aids in text string analysis.
An n-gram transformation takes a text variable as input and outputs strings by sliding a window of size *n* words, where *n* is specified by the user, over the text, and outputting every string of words of size *n* and all smaller sizes. For example,

specifying the n-gram transformation with window size =2 returns all the two-word combinations and all of the single words.

node

Amazon Elasticsearch Service (Amazon ES) (p. 191): An Elasticsearch instance. A node can be either a data instance or a dedicated master instance.
See Also dedicated master node.

NoEcho

A property of AWS CloudFormation (p. 195) parameters that prevent the otherwise default reporting of names and values of a template parameter. Declaring the `NoEcho` property causes the parameter value to be masked with asterisks in the report by the `cfn-describe-stacks` command.

NoSQL

Nonrelational database systems that are highly available, scalable, and optimized for high performance. Instead of the relational model, NoSQL databases (like Amazon DynamoDB (p. 190)) use alternate models for data management, such as key–value pairs or document storage.

null object

A null object is one whose version ID is null. Amazon S3 (p. 193) adds a null object to a bucket (p. 200) when versioning (p. 235) for that bucket is suspended. It is possible to have only one null object for each key in a bucket.

number of passes

The number of times that you allow Amazon Machine Learning to use the same data records to train a machine learning model.

O

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

object

Amazon Simple Storage Service (Amazon S3) (p. 193): The fundamental entity type stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3.

Amazon CloudFront (p. 190): Any entity that can be served either over HTTP or a version of RTMP.

observation

Amazon Machine Learning: A single instance of data that Amazon Machine Learning (Amazon ML) uses to either train a machine learning model how to predict or to generate a prediction. Each row in an Amazon ML input data file is an observation.

On-Demand Instance

An Amazon EC2 (p. 191) pricing option that charges you for compute capacity by the hour with no long-term commitment.

operation

An API function. Also called an *action*.

optimistic locking

A strategy to ensure that an item that you want to update has not been modified by others before you perform the update. For Amazon DynamoDB (p. 190), optimistic locking support is provided by the AWS SDKs.

organization

AWS Organizations (p. 198): An entity that you create to consolidate and manage your AWS accounts. An organization has one master account along with zero or more member accounts.

organizational unit

AWS Organizations (p. 198): A container for accounts within a root (p. 226) of an organization. An organizational unit (OU) can contain other OUs.

origin access identity

Also called OAI. When using Amazon CloudFront (p. 190) to serve content with an Amazon S3 (p. 193) bucket (p. 200) as the origin, a virtual identity that you use to require users to access your content through CloudFront URLs instead of Amazon S3 URLs. Usually used with CloudFront private content (p. 222).

origin server

The Amazon S3 (p. 193) bucket (p. 200) or custom origin containing the definitive original version of the content you deliver through CloudFront (p. 190).

original environment

The instances in a deployment group at the start of an AWS CodeDeploy blue/green deployment.

OSB transformation

Orthogonal sparse bigram transformation. In machine learning, a transformation that aids in text string analysis and that is an alternative to the n-gram transformation. OSB transformations are generated by sliding the window of size *n* words over the text, and outputting every pair of words that includes the first word in the window.
See Also n-gram transformation.

OU

See organizational unit.

output location

Amazon Machine Learning: An Amazon S3 location where the results of a batch prediction are stored.

P

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

pagination

The process of responding to an API request by returning a large list of records in small separate parts. Pagination can occur in the following situations:

- The client sets the maximum number of returned records to a value below the total number of records.
- The service has a default maximum number of returned records that is lower than the total number of records.

When an API response is paginated, the service sends a subset of the large list of records and a pagination token that indicates that more records are available. The client includes this pagination token in a subsequent API request, and the service responds with the next subset of records. This continues until the service responds with a subset of records and no pagination token, indicating that all records have been sent.

pagination token

A marker that indicates that an API response contains a subset of a larger list of records. The client can return this marker in a subsequent API request to retrieve the next subset of records until the service responds with a subset of records and no pagination token, indicating that all records have been sent.
See Also pagination.

paid AMI

An Amazon Machine Image (AMI) (p. 192) that you sell to other Amazon EC2 (p. 191) users on AWS Marketplace (p. 197).

paravirtual virtualization

See PV virtualization.

part

A contiguous portion of the object's data in a multipart upload request.

partition key

A simple primary key, composed of one attribute (also known as a *hash attribute*). See Also partition key, sort key.

PAT

Port address translation.

pebibyte

A contraction of peta binary byte, a pebibyte is 2^50 or 1,125,899,906,842,624 bytes. A petabyte (PB) is 10^15 or 1,000,000,000,000,000 bytes. 1,024 PiB is an exbibyte (p. 209).

period

See sampling period.

permission

A statement within a policy (p. 221) that allows or denies access to a particular resource (p. 225). You can state any permission like this: "A has permission to do B to C." For example, Jane (A) has permission to read messages (B) from John's Amazon SQS (p. 193) queue (C). Whenever Jane sends a request to Amazon SQS to use John's queue, the service checks to see if she has permission and if the request satisfies the conditions John set forth in the permission.

persistent storage

A data storage solution where the data remains intact until it is deleted. Options within AWS (p. 193) include: Amazon S3 (p. 193), Amazon RDS (p. 192), Amazon DynamoDB (p. 190), and other services.

physical name

A unique label that AWS CloudFormation (p. 195) assigns to each resource (p. 225) when creating a stack (p. 230). Some AWS CloudFormation commands accept the physical name as a value with the `--physical-name` parameter.

pipeline

AWS CodePipeline (p. 196): A workflow construct that defines the way software changes go through a release process.

plaintext

Information that has not been encrypted (p. 208), as opposed to ciphertext (p. 201).

policy

IAM (p. 197): A document defining permissions that apply to a user, group, or role; the permissions in turn determine what users can do in AWS. A policy typically allow (p. 189)s access to specific actions, and can optionally grant that the actions are allowed for specific resource (p. 225)s, like EC2 instance (p. 207)s, Amazon S3 (p. 193) bucket (p. 200)s, and so on. Policies can also explicitly deny (p. 206) access.

Auto Scaling (p. 195): An object that stores the information needed to launch or terminate instances for an Auto Scaling group. Executing the policy causes instances to be launched or terminated. You can configure an alarm (p. 189) to invoke an Auto Scaling policy.

policy generator

A tool in the IAM (p. 197) AWS Management Console (p. 197) that helps you build a policy (p. 221) by selecting elements from lists of available options.

policy simulator

A tool in the IAM (p. 197) AWS Management Console (p. 197) that helps you test and troubleshoot policies (p. 221) so you can see their effects in real-world scenarios.

policy validator

A tool in the IAM (p. 197) AWS Management Console (p. 197) that examines your existing IAM access control policies (p. 221) to ensure that they comply with the IAM policy grammar.

presigned URL

A web address that uses query string authentication (p. 223).

prefix

See job prefix.

Premium Support

A one-on-one, fast-response support channel that AWS customers can subscribe to for support for AWS infrastructure services.
See Also https://aws.amazon.com/premiumsupport/.

primary key

One or two attributes that uniquely identify each item in a Amazon DynamoDB (p. 190) table, so that no two items can have the same key.
See Also partition key, sort key.

primary shard

See shard.

principal

The user (p. 234), service, or account (p. 189) that receives permissions that are defined in a policy (p. 221). The principal is A in the statement "A has permission to do B to C."

private content

When using Amazon CloudFront (p. 190) to serve content with an Amazon S3 (p. 193) bucket (p. 200) as the origin, a method of controlling access to your content by requiring users to use signed URLs. Signed URLs can restrict user access based on the current date and time and/or the IP addresses that the requests originate from.

private IP address

A private numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). All EC2 instance (p. 207)ss are assigned two IP addresses at launch, which are directly mapped to each other through Network Address Translation (NAT (p. 218)): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC (p. 193) are assigned only a private IP address.

private subnet

A VPC (p. 235) subnet (p. 231) whose instances cannot be reached from the Internet.

product code

An identifier provided by AWS when you submit a product to AWS Marketplace (p. 197).

properties

See resource property.

property rule

A JSON (p. 214)-compliant markup standard for declaring properties, mappings, and output values in an AWS CloudFormation (p. 195) template.

Provisioned IOPS

A storage option designed to deliver fast, predictable, and consistent I/O performance. When you specify an IOPS rate while creating a DB instance, Amazon RDS (p. 192) provisions that IOPS rate for the lifetime of the DB instance.

pseudo parameter

A predefined setting, such as `AWS:StackName` that can be used in AWS CloudFormation (p. 195) templates without having to declare them. You can use pseudo parameters anywhere you can use a regular parameter.

public AMI

An Amazon Machine Image (AMI) (p. 192) that all AWS account (p. 189)s have permission to launch.

public data set

A large collection of public information that can be seamlessly integrated into AWS cloud-based applications. Amazon stores public data sets at no charge to the community and, like all AWS services, users pay only for the compute and storage they use for their own applications. These data sets currently include data from the Human Genome Project, the U.S. Census, Wikipedia, and other sources.
See Also https://aws.amazon.com/publicdatasets.

public IP address

A pubic numerical address (for example, 192.0.2.44) that networked devices use to communicate with one another using the Internet Protocol (IP). EC2 instance (p. 207)s are assigned two IP addresses at launch, which are directly mapped to each other through Network Address Translation (NAT (p. 218)): a private address (following RFC 1918) and a public address. *Exception:* Instances launched in Amazon VPC (p. 193) are assigned only a private IP address.

public subnet

A subnet (p. 231) whose instances can be reached from the Internet.

PV virtualization

Paravirtual virtualization. Allows guest VMs to run on host systems that do not have special support extensions for full hardware and CPU virtualization. Because PV guests run a modified operating system that does not use hardware emulation, they cannot provide hardware-related features such as enhanced networking or GPU support.
See Also HVM virtualization.

Q

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

quartile binning transformation

Amazon Machine Learning: A process that takes two inputs, a numerical variable and a parameter called a bin number, and outputs a categorical variable. Quartile binning transformations discover non-linearity in a variable's distribution by enabling the machine learning model to learn separate importance values for parts of the numeric variable’s distribution.

Query

A type of web service that generally uses only the GET or POST HTTP method and a query string with parameters in the URL.
See Also REST.

query string authentication

An AWS feature that lets you place the authentication information in the HTTP request query string instead of in the `Authorization` header, which enables URL-based access to objects in a bucket (p. 200).

queue

A sequence of messages or jobs that are held in temporary storage awaiting transmission or processing.

queue URL

A web address that uniquely identifies a queue.

quota

Amazon RDS (p. 192): The maximum number of DB instance (p. 205)s and available storage you can use.

Amazon ElastiCache (p. 191): The maximum number of the following items:

- The number of cache clusters for each AWS account (p. 189)
- The number of cache nodes per cache cluster
- The total number of cache nodes per AWS account across all cache clusters created by that AWS account

R

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

range GET

A request that specifies a byte range of data to get for a download. If an object is large, you can break up a download into smaller units by sending multiple range GET requests that each specify a different byte range to GET.

raw email

A type of *sendmail* request with which you can specify the email headers and MIME types.

RDS

See Amazon Relational Database Service (Amazon RDS).

read replica

Amazon RDS (p. 192): An active copy of another DB instance. Any updates to the data on the source DB instance are replicated to the read replica DB instance using the built-in replication feature of MySQL 5.1.

real-time predictions

Amazon Machine Learning: Synchronously generated predictions for individual data observations.
See Also batch prediction.

receipt handle

Amazon SQS (p. 193): An identifier that you get when you receive a message from the queue. This identifier is required to delete a message from the queue or when changing a message's visibility timeout.

receiver

The entity that consists of the network systems, software, and policies that manage email delivery for a recipient (p. 224).

recipient

Amazon Simple Email Service (Amazon SES) (p. 193): The person or entity receiving an email message. For example, a person named in the "To" field of a message.

Redis

A fast, open source, in-memory key-value data structure store. Redis comes with a set of versatile in-memory data structures with which you can easily create a variety of custom applications.

reference

A means of inserting a property from one AWS resource (p. 225) into another. For example, you could insert an Amazon EC2 (p. 191) security group (p. 228) property into an Amazon RDS (p. 192) resource.

region

A named set of AWS resource (p. 225)s in the same geographical area. A region comprises at least two Availability Zone (p. 195)s.

regression model

Amazon Machine Learning: Preformatted instructions for common data transformations that fine-tune machine learning model performance.

regression model

A type of machine learning model that predicts a numeric value, such as the exact purchase price of a house.

regularization

A machine learning (ML) parameter that you can tune to obtain higher-quality ML models. Regularization helps prevent ML models from memorizing training data examples instead of learning how to generalize the patterns it sees (called overfitting). When training data is overfitted, the ML model performs well on the training data but does not perform well on the evaluation data or on new data.

replacement environment

The instances in a deployment group after the AWS CodeDeploy blue/green deployment.

replica shard

See shard.

reply path

The email address to which an email reply is sent. This is different from the return path (p. 226).

representational state transfer

See REST.

reputation

1. An Amazon SES (p. 193) metric, based on factors that might include bounce (p. 200)s, complaint (p. 202)s, and other metrics, regarding whether or not a customer is sending high-quality email.

2. A measure of confidence, as judged by an Internet service provider (p. 213) or other entity that an IP address that they are receiving email from is not the source of spam (p. 230).

requester

The person (or application) that sends a request to AWS to perform a specific action. When AWS receives a request, it first evaluates the requester's permissions to determine whether the requester is allowed to perform the request action (if applicable, for the requested resource (p. 225)).

Requester Pays

An Amazon S3 (p. 193) feature that allows a bucket owner (p. 200) to specify that anyone who requests access to objects in a particular bucket (p. 200) must pay the data transfer and request costs.

reservation

A collection of EC2 instance (p. 207)s started as part of the same launch request. Not to be confused with a Reserved Instance (p. 225).

Reserved Instance

A pricing option for EC2 instance (p. 207)s that discounts the on-demand (p. 219) usage charge for instances that meet the specified parameters. Customers pay for the entire term of the instance, regardless of how they use it.

Reserved Instance Marketplace

An online exchange that matches sellers who have reserved capacity that they no longer need with buyers who are looking to purchase additional capacity. Reserved Instance (p. 225)s that you purchase from third-party sellers have less than a full standard term remaining and can be sold at different upfront prices. The usage or reoccurring fees remain the same as the fees set when the Reserved Instances were originally purchased. Full standard terms for Reserved Instances available from AWS run for one year or three years.

resource

An entity that users can work with in AWS, such as an EC2 instance (p. 207), a Amazon DynamoDB (p. 190) table, an Amazon S3 (p. 193) bucket (p. 200), an IAM (p. 197) user, an AWS OpsWorks (p. 197) stack (p. 230), and so on.

resource property

A value required when including an AWS resource (p. 225) in an AWS CloudFormation (p. 195) stack (p. 230). Each resource may have one or more properties associated with it. For example, an `AWS::EC2::Instance` resource may have a `UserData` property. In an AWS CloudFormation template, resources must declare a properties section, even if the resource has no properties.

resource record

Also called *resource record set*. The fundamental information elements in the Domain Name System (DNS).
See Also Domain Name System in Wikipedia.

REST

Representational state transfer. A simple stateless architecture that generally runs over HTTPS/TLS. REST emphasizes that resources have unique and hierarchical identifiers (URIs), are represented by common media types (HTML, XML, JSON (p. 214), and so on), and that operations on the resources are either predefined or discoverable within the media type. In practice, this generally results in a limited number of operations.
See Also Query, WSDL, SOAP.

RESTful web service

Also known as RESTful API. A web service that follows REST (p. 225) architectural constraints. The API operations must use HTTP methods explicitly; expose hierarchical URIs; and transfer either XML, JSON (p. 214), or both.

HTTP-Query

See Query.

return enabled

Amazon CloudSearch (p. 190): An index field option that enables the field's values to be returned in the search results.

return path

The email address to which bounced email is returned. The return path is specified in the header of the original email. This is different from the reply path (p. 224).

revision

AWS CodePipeline (p. 196): A change made to a source that is configured in a source action, such as a pushed commit to a GitHub (p. 211) repository or an update to a file in a versioned Amazon S3 (p. 193) bucket (p. 200).

role

A tool for giving temporary access to AWS resource (p. 225)s in your AWS account (p. 189).

rollback

A return to a previous state that follows the failure to create an object, such as AWS CloudFormation (p. 195) stack (p. 230). All resource (p. 225)s associated with the failure are deleted during the rollback. For AWS CloudFormation, you can override this behavior using the `--disable-rollback` option on the command line.

root

AWS Organizations (p. 198): A parent container for the accounts in your organization. If you apply a service control policy (p. 228) to the root, it applies to every organizational unit (p. 219) and account in the organization.

root credentials

Authentication information associated with the AWS account (p. 189) owner.

root device volume

A volume (p. 235) that contains the image used to boot the instance (p. 213). If you launched the instance from an AMI (p. 192) backed by instance store (p. 213), this is an instance store volume (p. 235) created from a template stored in Amazon S3 (p. 193). If you launched the instance from an AMI backed by Amazon EBS (p. 190), this is an Amazon EBS volume created from an Amazon EBS snapshot.

route table

A set of routing rules that controls the traffic leaving any subnet (p. 231) that is associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.

row identifier

row ID.Amazon Machine Learning: An attribute in the input data that you can include in the evaluation or prediction output to make it easier to associate a prediction with an observation.

rule

AWS WAF (p. 199): A set of conditions that AWS WAF searches for in web requests to AWS resource (p. 225)s such as Amazon CloudFront (p. 190) distributions. You add rules to a web ACL (p. 236), and then specify whether you want to allow or block web requests based on each rule.

S

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

S3

See Amazon Simple Storage Service (Amazon S3).

sampling period

A defined duration of time, such as one minute, over which Amazon CloudWatch (p. 190) computes a statistic (p. 230).

sandbox

A testing location where you can test the functionality of your application without affecting production, incurring charges, or purchasing products.

Amazon SES (p. 193): An environment that is designed for developers to test and evaluate the service. In the sandbox, you have full access to the Amazon SES API, but you can only send messages to verified email addresses and the mailbox simulator. To get out of the sandbox, you need to apply for production access. Accounts in the sandbox also have lower sending limits (p. 228) than production accounts.

scale in

To remove EC2 instances from an Auto Scaling group (p. 195).

scale out

To add EC2 instances to an Auto Scaling group (p. 195).

scaling policy

A description of how Auto Scaling should automatically scale an Auto Scaling group (p. 195) in response to changing demand.
See Also scale in, scale out.

scaling activity

A process that changes the size, configuration, or makeup of an Auto Scaling group (p. 195) by launching or terminating instances.

scheduler

The method used for placing task (p. 233)s on container instance (p. 203)s.

schema

Amazon Machine Learning: The information needed to interpret the input data for a machine learning model, including attribute names and their assigned data types, and the names of special attributes.

score cut-off value

Amazon Machine Learning: A binary classification models output a score that ranges from 0 to 1. To decide whether an observation should be classified as 1 or 0, you pick a classification threshold, or cut-off, and Amazon ML compares the score against it. Observations with scores higher than the cut-off are predicted as target equals 1, and scores lower than the cut-off are predicted as target equals 0.

SCP

See service control policy.

search API

Amazon CloudSearch (p. 190): The API that you use to submit search requests to a search domain (p. 227).

search domain

Amazon CloudSearch (p. 190): Encapsulates your searchable data and the search instances that handle your search requests. You typically set up a separate Amazon CloudSearch domain for each different collection of data that you want to search.

search domain configuration

Amazon CloudSearch (p. 190): An domain's indexing options, analysis scheme (p. 194)s, expression (p. 209)s, suggester (p. 232)s, access policies, and scaling and availability options.

search enabled

Amazon CloudSearch (p. 190): An index field option that enables the field data to be searched.

search endpoint

Amazon CloudSearch (p. 190): The URL that you connect to when sending search requests to a search domain. Each Amazon CloudSearch domain has a unique search endpoint that remains the same for the life of the domain.

search index

Amazon CloudSearch (p. 190): A representation of your searchable data that facilitates fast and accurate data retrieval.

search instance

Amazon CloudSearch (p. 190): A compute resource (p. 225) that indexes your data and processes search requests. An Amazon CloudSearch domain has one or more search instances, each with a finite amount of RAM and CPU resources. As your data volume grows, more search instances or larger search instances are deployed to contain your indexed data. When necessary, your index is automatically partitioned across multiple search instances. As your request volume or complexity increases, each search partition is automatically replicated to provide additional processing capacity.

search request

Amazon CloudSearch (p. 190): A request that is sent to an Amazon CloudSearch domain's search endpoint to retrieve documents from the index that match particular search criteria.

search result

Amazon CloudSearch (p. 190): A document that matches a search request. Also referred to as a *search hit*.

secret access key

A key that is used in conjunction with the access key ID (p. 188) to cryptographically sign programmatic AWS requests. Signing a request identifies the sender and prevents the request from being altered. You can generate secret access keys for your AWS account (p. 189), individual IAM user (p. 234)s, and temporary sessions.

security group

A named set of allowed inbound network connections for an instance. (Security groups in Amazon VPC (p. 193) also include support for outbound connections.) Each security group consists of a list of protocols, ports, and IP address ranges. A security group can apply to multiple instances, and multiple groups can regulate a single instance.

sender

The person or entity sending an email message.

Sender ID

A Microsoft-controlled version of SPF (p. 230). An email authentication and anti-spoofing system. For more information about Sender ID, see Sender ID in Wikipedia.

sending limits

The sending quota (p. 228) and maximum send rate (p. 217) that are associated with every Amazon SES (p. 193) account.

sending quota

The maximum number of email messages that you can send using Amazon SES (p. 193) in a 24-hour period.

server-side encryption (SSE)

The encrypting (p. 208) of data at the server level. Amazon S3 (p. 193) supports three modes of server-side encryption: SSE-S3, in which Amazon S3 manages the keys; SSE-C, in which the customer manages the keys; and SSE-KMS, in which AWS Key Management Service (AWS KMS) (p. 197) manages keys.

service

See Amazon ECS service.

service control policy

AWS Organizations (p. 198): A policy-based control that specifies the services and actions that users and roles can use in the accounts that the service control policy (SCP) affects.

service endpoint

See endpoint.

service health dashboard

A web page showing up-to-the-minute information about AWS service availability. The dashboard is located at http://status.aws.amazon.com/.

service role

An IAM (p. 197) role (p. 226) that grants permissions to an AWS service so it can access AWS resource (p. 225)s. The policies that you attach to the service role determine which AWS resources the service can access and what it can do with those resources.

SES

See Amazon Simple Email Service (Amazon SES).

session

The period during which the temporary security credentials provided by AWS Security Token Service (AWS STS) (p. 198) allow access to your AWS account.

SHA

Secure Hash Algorithm. SHA1 is an earlier version of the algorithm, which AWS has deprecated in favor of SHA256.

shard

Amazon Elasticsearch Service (Amazon ES) (p. 191): A partition of data in an index. You can split an index into multiple shards, which can include primary shards (original shards) and replica shards (copies of the primary shards). Replica shards provide failover, which means that a replica shard is promoted to a primary shard if a cluster node that contains a primary shard fails. Replica shards also can handle requests.

shared AMI

An Amazon Machine Image (AMI) (p. 192) that a developer builds and makes available for others to use.

shutdown action

Amazon EMR (p. 191): A predefined bootstrap action that launches a script that executes a series of commands in parallel before terminating the job flow.

signature

Refers to a *digital signature*, which is a mathematical way to confirm the authenticity of a digital message. AWS uses signatures to authenticate the requests you send to our web services. For more information, to https://aws.amazon.com/security.

SIGNATURE file

AWS Import/Export (p. 197): A file you copy to the root directory of your storage device. The file contains a job ID, manifest file, and a signature.

Signature Version 4

Protocol for authenticating inbound API requests to AWS services in all AWS regions.

Simple Mail Transfer Protocol

See SMTP.

Simple Object Access Protocol

See SOAP.

Simple Storage Service

See Amazon Simple Storage Service (Amazon S3).

Single-AZ DB instance

A standard (non-Multi-AZ) DB instance (p. 205) that is deployed in one Availability Zone (p. 195), without a standby replica in another Availability Zone.
See Also Multi-AZ deployment.

sloppy phrase search

A search for a phrase that specifies how close the terms must be to one another to be considered a match.

SMTP

Simple Mail Transfer Protocol. The standard that is used to exchange email messages between Internet hosts for the purpose of routing and delivery.

snapshot

Amazon Elastic Block Store (Amazon EBS) (p. 190): A backup of your volume (p. 235)s that is stored in Amazon S3 (p. 193). You can use these snapshots as the starting point for new Amazon EBS volumes or to protect your data for long-term durability.
See Also DB snapshot.

SNS

See Amazon Simple Notification Service (Amazon SNS).

Snowball

An AWS Import/Export (p. 197) feature that uses Amazon-owned Snowball appliances for transferring your data.
See Also https://aws.amazon.com/importexport.

SOAP

Simple Object Access Protocol. An XML-based protocol that lets you exchange information over a particular protocol (HTTP or SMTP, for example) between applications.
See Also REST, WSDL.

soft bounce

A temporary email delivery failure such as one resulting from a full mailbox.

software VPN

A software appliance-based VPN connection over the Internet.

sort enabled

Amazon CloudSearch (p. 190): An index field option that enables a field to be used to sort the search results.

sort key

An attribute used to sort the order of partition keys in a composite primary key (also known as a *range attribute*).
See Also partition key, primary key.

source/destination checking

A security measure to verify that an EC2 instance (p. 207) is the origin of all traffic that it sends and the ultimate destination of all traffic that it receives; that is, that the instance is not relaying traffic. Source/destination checking is enabled by default. For instances that function as gateways, such as VPC (p. 235) NAT (p. 218) instances, source/destination checking must be disabled.

spam

Unsolicited bulk email.

spamtrap

An email address that is set up by an anti-spam (p. 230) entity, not for correspondence, but to monitor unsolicited email. This is also called a *honeypot*.

SPF

Sender Policy Framework. A standard for authenticating email.
See Also http://www.openspf.org.

Spot Instance

A type of EC2 instance (p. 207) that you can bid on to take advantage of unused Amazon EC2 (p. 191) capacity.

Spot price

The price for a Spot Instance (p. 230) at any given time. If your maximum price exceeds the current price and your restrictions are met, Amazon EC2 (p. 191) launches instances on your behalf.

SQL injection match condition

AWS WAF (p. 199): An attribute that specifies the part of web requests, such as a header or a query string, that AWS WAF inspects for malicious SQL code. Based on the specified conditions, you can configure AWS WAF to allow or block web requests to AWS resource (p. 225)s such as Amazon CloudFront (p. 190) distributions.

SQS

See Amazon Simple Queue Service (Amazon SQS).

SSE

See server-side encryption (SSE).

SSL

Secure Sockets Layer
See Also Transport Layer Security.

stack

AWS CloudFormation (p. 195): A collection of AWS resource (p. 225)s that you create and delete as a single unit.

AWS OpsWorks (p. 197): A set of instances that you manage collectively, typically because they have a common purpose such as serving PHP applications. A stack serves as a container and handles tasks that apply to the group of instances as a whole, such as managing applications and cookbooks.

station

AWS CodePipeline (p. 196): A portion of a pipeline workflow where one or more actions are performed.

station

A place at an AWS facility where your AWS Import/Export data is transferred on to, or off of, your storage device.

statistic

One of five functions of the values submitted for a given sampling period (p. 226). These functions are `Maximum`, `Minimum`, `Sum`, `Average`, and `SampleCount`.

stem

The common root or substring shared by a set of related words.

stemming

The process of mapping related words to a common stem. This enables matching on variants of a word. For example, a search for "horse" could return matches for horses, horseback, and horsing, as well as horse. Amazon CloudSearch (p. 190) supports both dictionary based and algorithmic stemming.

step

Amazon EMR (p. 191): A single function applied to the data in a job flow (p. 214). The sum of all steps comprises a job flow.

step type

Amazon EMR (p. 191): The type of work done in a step. There are a limited number of step types, such as moving data from Amazon S3 (p. 193) to Amazon EC2 (p. 191) or from Amazon EC2 to Amazon S3.

sticky session

A feature of the Elastic Load Balancing (p. 207) load balancer that binds a user's session to a specific application instance so that all requests coming from the user during the session are sent to the same application instance. By contrast, a load balancer defaults to route each request independently to the application instance with the smallest load.

stopping

The process of filtering stop words from an index or search request.

stopword

A word that is not indexed and is automatically filtered out of search requests because it is either insignificant or so common that including it would result in too many matches to be useful. Stop words are language-specific.

streaming

Amazon EMR (Amazon EMR) (p. 191): A utility that comes with Hadoop (p. 211) that enables you to develop MapReduce executables in languages other than Java.

Amazon CloudFront (p. 190): The ability to use a media file in real time—as it is transmitted in a steady stream from a server.

streaming distribution

A special kind of distribution (p. 206) that serves streamed media files using a Real Time Messaging Protocol (RTMP) connection.

Streams

See Amazon Kinesis Streams.

string-to-sign

Before you calculate an HMAC (p. 212) signature, you first assemble the required components in a canonical order. The preencrypted string is the string-to-sign.

string match condition

AWS WAF (p. 199): An attribute that specifies the strings that AWS WAF searches for in a web request, such as a value in a header or a query string. Based on the specified strings, you can configure AWS WAF to allow or block web requests to AWS resource (p. 225)s such as CloudFront (p. 190) distributions.

strongly consistent read

A read process that returns a response with the most up-to-date data, reflecting the updates from all prior write operations that were successful—regardless of the region.
See Also data consistency, eventual consistency, eventually consistent read.

structured query

Search criteria specified using the Amazon CloudSearch (p. 190) structured query language. You use the structured query language to construct compound queries that use advanced search options and combine multiple search criteria using Boolean operators.

STS

See AWS Security Token Service (AWS STS).

subnet

A segment of the IP address range of a VPC (p. 235) that EC2 instance (p. 207)s can be attached to. You can create subnets to group instances according to security and operational needs.

Subscription button

An HTML-coded button that enables an easy way to charge customers a recurring fee.

suggester

Amazon CloudSearch (p. 190): Specifies an index field you want to use to get autocomplete suggestions and options that can enable fuzzy matches and control how suggestions are sorted.

suggestions

Documents that contain a match for the partial search string in the field designated by the suggester (p. 232). Amazon CloudSearch (p. 190) suggestions include the document IDs and field values for each matching document. To be a match, the string must match the contents of the field starting from the beginning of the field.

supported AMI

An Amazon Machine Image (AMI) (p. 192) similar to a paid AMI (p. 220), except that the owner charges for additional software or a service that customers use with their own AMIs.

SWF

See Amazon Simple Workflow Service (Amazon SWF).

symmetric encryption

Encryption (p. 208) that uses a private key only.
See Also asymmetric encryption.

synchronous bounce

A type of bounce (p. 200) that occurs while the email servers of the sender (p. 228) and receiver (p. 224) are actively communicating.

synonym

A word that is the same or nearly the same as an indexed word and that should produce the same results when specified in a search request. For example, a search for "Rocky Four" or "Rocky 4" should return the fourth *Rocky* movie. This can be done by designating that `four` and `4` are synonyms for `IV`. Synonyms are language-specific.

T

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

table

A collection of data. Similar to other database systems, DynamoDB stores data in tables.

tag

Metadata that you can define and assign to AWS resource (p. 225)s, such as an EC2 instance (p. 207). Not all AWS resources can be tagged.

tagging

Tagging resources: Applying a tag (p. 232) to an AWS resource (p. 225).

Amazon SES (p. 193): Also called *labeling*. A way to format return path (p. 226) email addresses so that you can specify a different return path for each recipient of a message. Tagging enables you to support VERP (p. 235). For example, if Andrew manages a mailing list, he can use the return paths andrew +recipient1@example.net and andrew+recipient2@example.net so that he can determine which email bounced.

target attribute

Amazon Machine Learning (Amazon ML): The attribute in the input data that contains the “correct” answers. Amazon ML uses the target attribute to learn how to make predictions on new data. For example, if you were building a model for predicting the sale price of a house, the target attribute would be “target sale price in USD.”

target revision

AWS CodeDeploy (p. 196): The most recent version of the application revision that has been uploaded to the repository and will be deployed to the instances in a deployment group. In other words, the application revision currently targeted for deployment. This is also the revision that will be pulled for automatic deployments.

task

An instantiation of a task definition (p. 233) that is running on a container instance (p. 203).

task definition

The blueprint for your task. Specifies the name of the task (p. 233), revisions, container definition (p. 203)s, and volume (p. 235) information.

task node

An EC2 instance (p. 207) that runs Hadoop (p. 211) map and reduce tasks, but does not store data. Task nodes are managed by the master node (p. 216), which assigns Hadoop tasks to nodes and monitors their status. While a job flow is running you can increase and decrease the number of task nodes. Because they don't store data and can be added and removed from a job flow, you can use task nodes to manage the EC2 instance capacity your job flow uses, increasing capacity to handle peak loads and decreasing it later.

Task nodes only run a TaskTracker Hadoop daemon.

tebibyte

A contraction of tera binary byte, a tebibyte is 2^40 or 1,099,511,627,776 bytes. A terabyte (TB) is 10^12 or 1,000,000,000,000 bytes. 1,024 TiB is a pebibyte (p. 221).

template format version

The version of an AWS CloudFormation (p. 195) template design that determines the available features. If you omit the `AWSTemplateFormatVersion` section from your template, AWS CloudFormation assumes the most recent format version.

template validation

The process of confirming the use of JSON (p. 214) code in an AWS CloudFormation (p. 195) template. You can validate any AWS CloudFormation template using the `cfn-validate-template` command.

temporary security credentials

Authentication information that is provided by AWS STS (p. 198) when you call an STS API action. Includes an access key ID (p. 188), a secret access key (p. 228), a session (p. 228) token, and an expiration time.

throttling

The automatic restricting or slowing down of a process based on one or more limits. Examples: Amazon Kinesis Streams (p. 192) throttles operations if an application (or group of applications operating on the same stream) attempts to get data from a shard at a rate faster than the shard limit. Amazon API Gateway (p. 189) uses throttling to limit the steady-state request rates for a single account. Amazon SES (p. 193) uses throttling to reject attempts to send email that exceeds the sending limits (p. 228).

time series data

Data provided as part of a metric. The time value is assumed to be when the value occurred. A metric is the fundamental concept for Amazon CloudWatch (p. 190) and represents a time-ordered set of data points. You publish metric data points into CloudWatch and later retrieve statistics about those data points as a time-series ordered data set.

time stamp

A date/time string in ISO 8601 format.

TLS

See Transport Layer Security.

tokenization

The process of splitting a stream of text into separate tokens on detectable boundaries such as whitespace and hyphens.

topic

A communication channel to send messages and subscribe to notifications. It provides an access point for publishers and subscribers to communicate with each other.

training datasource

A datasource that contains the data that Amazon Machine Learning uses to train the machine learning model to make predictions.

transition

AWS CodePipeline (p. 196): The act of a revision in a pipeline continuing from one stage to the next in a workflow.

Transport Layer Security

A cryptographic protocol that provides security for communication over the Internet. Its predecessor is Secure Sockets Layer (SSL).

trust policy

An IAM (p. 197) policy (p. 221) that is an inherent part of an IAM role (p. 226). The trust policy specifies which principal (p. 222)s are allowed to use the role.

trusted signers

AWS account (p. 189)s that the CloudFront (p. 190) distribution owner has given permission to create signed URLs for a distribution's content.

tuning

Selecting the number and type of AMIs (p. 192) to run a Hadoop (p. 211) job flow most efficiently.

tunnel

A route for transmission of private network traffic that uses the Internet to connect nodes in the private network. The tunnel uses encryption and secure protocols such as PPTP to prevent the traffic from being intercepted as it passes through public routing nodes.

U

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

unbounded

The number of potential occurrences is not limited by a set number. This value is often used when defining a data type that is a list (for example, `maxOccurs="unbounded"`), in WSDL (p. 236).

unit

Standard measurement for the values submitted to Amazon CloudWatch (p. 190) as metric data. Units include seconds, percent, bytes, bits, count, bytes/second, bits/second, count/second, and none.

unlink from VPC

The process of unlinking (or detaching) an EC2-Classic instance (p. 213) from a ClassicLink-enabled VPC (p. 235).
See Also ClassicLink, link to VPC.

usage report

An AWS record that details your usage of a particular AWS service. You can generate and download usage reports from https://aws.amazon.com/usage-reports/.

user

A person or application under an account (p. 189) that needs to make API calls to AWS products. Each user has a unique name within the AWS account, and a set of security credentials not shared with other users. These credentials are separate from the AWS account's security credentials. Each user is associated with one and only one AWS account.

V

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

validation

See template validation.

value

Instances of attributes (p. 195) for an item, such as cells in a spreadsheet. An attribute might have multiple values.

Tagging resources: A specific tag (p. 232) label that acts as a descriptor within a tag category (key). For example, you might have EC2 instance (p. 207) with the tag key of *Owner* and the tag value of *Jan*. You can tag an AWS resource (p. 225) with up to 10 key–value pairs. Not all AWS resources can be tagged.

Variable Envelope Return Path

See VERP.

verification

The process of confirming that you own an email address or a domain so that you can send email from or to it.

VERP

Variable Envelope Return Path. A way in which email sending applications can match bounce (p. 200)d email with the undeliverable address that caused the bounce by using a different return path (p. 226) for each recipient. VERP is typically used for mailing lists. With VERP, the recipient's email address is embedded in the address of the return path, which is where bounced email is returned. This makes it possible to automate the processing of bounced email without having to open the bounce messages, which may vary in content.

versioning

Every object in Amazon S3 (p. 193) has a key and a version ID. Objects with the same key, but different version IDs can be stored in the same bucket (p. 200). Versioning is enabled at the bucket layer using PUT Bucket versioning.

VGW

See virtual private gateway.

virtualization

Allows multiple guest virtual machines (VM) to run on a host operating system. Guest VMs can run on one or more levels above the host hardware, depending on the type of virtualization.
See Also PV virtualization, HVM virtualization.

virtual private cloud

See VPC.

virtual private gateway

(VGW) The Amazon side of a VPN connection (p. 236) that maintains connectivity. The internal interfaces of the virtual private gateway connect to your VPC (p. 235) via the VPN attachment and the external interfaces connect to the VPN connection, which leads to the customer gateway (p. 204).

visibility timeout

The period of time that a message is invisible to the rest of your application after an application component gets it from the queue. During the visibility timeout, the component that received the message usually processes it, and then deletes it from the queue. This prevents multiple components from processing the same message.

volume

A fixed amount of storage on an instance (p. 213). You can share volume data between container (p. 203)s and persist the data on the container instance (p. 203) when the containers are no longer running.

VPC

Virtual private cloud. An elastic network populated by infrastructure, platform, and application services that share common security and interconnection.

VPC endpoint

A feature that enables you to create a private connection between your VPC (p. 235) and an another AWS service without requiring access over the Internet, through a NAT (p. 218) instance, a VPN connection (p. 236), or AWS Direct Connect (p. 196).

VPG

See virtual private gateway.

VPN CloudHub

See AWS VPN CloudHub.

VPN connection

Amazon Web Services (AWS) (p. 193): The IPsec connection between a VPC (p. 235) and some other network, such as a corporate data center, home network, or co-location facility.

W

Numbers and Symbols (p. 188) | A (p. 188) | B (p. 199) | C (p. 200) | D (p. 204) | E (p. 207) | F (p. 209) | G (p. 210) | H (p. 211) | I (p. 212) | J (p. 214) | K (p. 214) | L (p. 215) | M (p. 216) | N (p. 218) | O (p. 219) | P (p. 220) | Q (p. 223) | R (p. 223) | S (p. 226) | T (p. 232) | U (p. 234) | V (p. 234) | W (p. 236) | X, Y, Z (p. 236)

WAM

See Amazon WorkSpaces Application Manager (Amazon WAM).

web access control list

AWS WAF (p. 199): A set of rules that defines the conditions that AWS WAF searches for in web requests to AWS resource (p. 225)s such as Amazon CloudFront (p. 190) distributions. A web access control list (web ACL) specifies whether to allow, block, or count the requests.

Web Services Description Language

See WSDL.

WSDL

Web Services Description Language. A language used to describe the actions that a web service can perform, along with the syntax of action requests and responses. See Also REST, SOAP.

X, Y, Z

X.509 certificate

An digital document that uses the X.509 public key infrastructure (PKI) standard to verify that a public key belongs to the entity described in the certificate (p. 201).

yobibyte

A contraction of yotta binary byte, a yobibyte is 2^80 or 1,208,925,819,614,629,174,706,176 bytes. A yottabyte (YB) is 10^24 or 1,000,000,000,000,000,000,000,000 bytes.

zebibyte

A contraction of zetta binary byte, a zebibyte is 2^70 or 1,180,591,620,717,411,303,424 bytes. A zettabyte (ZB) is 10^21 or 1,000,000,000,000,000,000,000 bytes. 1,024 ZiB is a yobibyte (p. 236).

zone awareness

Amazon Elasticsearch Service (Amazon ES) (p. 191): A configuration that distributes nodes in a cluster across two Availability Zone (p. 195)s in the same region. Zone awareness helps to prevent data loss and minimizes downtime in the event of node and data center failure. If you enable zone awareness, you must have an even number of data instances in the instance count, and you also must use the Amazon Elasticsearch Service Configuration API to replicate your data for your Elasticsearch cluster.