
AWS Support

User Guide

API Version 2013-04-15



AWS Support: User Guide

Copyright © 2017 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Getting Started with AWS Support	1
Features of AWS Support Plans	1
Case Management	2
Example: Creating a Case	3
Monitoring and Maintaining Your Case	5
Case History	6
Accessing AWS Support	6
AWS Account	6
IAM	6
AWS Trusted Advisor	7
Troubleshooting	7
Service-specific Troubleshooting	7
About the AWS Support API	10
Support Case Management	10
Trusted Advisor	11
Endpoint	11
Support in AWS SDKs	11
Programming the Life of an AWS Support Case	12
Overview	12
Using IAM with the AWS Support API	12
Create an AWS Support Client	13
Discover AWS Services and Issue Severity Levels	13
Create an Attachment Set	14
Create a Support Case	15
Retrieve and Update Support Case Communications	17
Retrieve All Support Case Information	19
Resolve a Support Case	20
Using Trusted Advisor as a Web Service	21
Get the List of Available Trusted Advisor Checks	21
Request a Trusted Advisor Check Result	22
Poll a Trusted Advisor Check for Status Changes	22
Print Details of a Trusted Advisor Check	23
Logging AWS Support API Calls with AWS CloudTrail	25
AWS Support Information in CloudTrail History	25
AWS Support Information in CloudTrail Logging	25
Understanding AWS Support Log File Entries	26
Monitoring Trusted Advisor Check Results with CloudWatch Events	28
Document History	30
AWS Glossary	31

Getting Started with AWS Support

AWS Support offers a range of plans that provide access to tools and expertise that support the success and operational health of your AWS solutions. All support plans provide 24x7 access to customer service, AWS documentation, whitepapers, and support forums. If you need technical support and more resources to help plan, deploy, and optimize your AWS environment, you can select a support plan that best aligns with your AWS use case.

Topics

- [Features of AWS Support Plans \(p. 1\)](#)
- [Case Management \(p. 2\)](#)
- [Accessing AWS Support \(p. 6\)](#)
- [AWS Trusted Advisor \(p. 7\)](#)
- [Troubleshooting \(p. 7\)](#)

Features of AWS Support Plans

AWS Support offers four support plans: Basic, Developer, Business, and Enterprise. The Basic plan is free of charge and offers support for account and billing questions and service limit increases. The other plans offer an unlimited number of technical support cases with pay-by-the-month pricing and no long-term contracts, providing the level of support that meets your needs.

All AWS customers automatically have around-the-clock access to these features of the Basic support plan:

- Customer Service: one-on-one responses to account and billing questions
- Support forums
- Service health checks
- Documentation, whitepapers, and best-practice guides

Customers with a Developer support plan have access to these additional features:

- Best-practice guidance
- Client-side diagnostic tools
- Building-block architecture support: guidance on how to use AWS products, features, and services together

In addition, customers with a Business or Enterprise support plan have access to these features:

- Use-case guidance: what AWS products, features, and services to use to best support your specific needs
- [AWS Identity and Access Management \(p. 6\)](#) (IAM) for controlling individuals' access to AWS Support
- [AWS Trusted Advisor \(p. 7\)](#), which inspects customer environments and identifies opportunities to save money, close security gaps, and improve system reliability and performance
- An API for interacting with Support Center and Trusted Advisor, allowing for automated support case management and Trusted Advisor operations
- Third-party software support: help with Amazon Elastic Compute Cloud (EC2) instance operating systems and configuration and performance of the most popular third-party software components on AWS

In addition, customers with an Enterprise support plan have access to these features:

- Application architecture guidance: consultative partnership supporting specific use cases and applications
- Infrastructure event management: short-term engagement with AWS Support to get a deep understanding of your use case and provide architectural and scaling guidance for an event
- AWS Concierge
- Technical account manager
- White-glove case routing
- Management business reviews

For more detailed information about features and pricing for each support plan, see [AWS Support](#) and [AWS Support Features](#). Some features, such as around-the-clock phone and chat support, are not available in all languages.

Case Management

You can sign in to the Support Center at <https://console.aws.amazon.com/support/home#/> by using the email address and password associated with your AWS account. To log in with other credentials, see [Accessing AWS Support \(p. 6\)](#).

There are three types of cases you can open:

- **Account and Billing Support** cases are available to all AWS customers. This case type connects you to customer service for help with billing and account-related questions.
- **Service Limit Increase** requests are also available to all AWS customers. For information on the default service limits, see [AWS Service Limits](#).
- **Technical Support** cases connect you to technical support for help with service-related technical issues and, in some cases, third-party applications. If you have a Developer support plan, you can

communicate via the web. If you have a Business or Enterprise support plan, you can also communicate by phone or live chat.

To open a Support case:

- In [Support Center](#), click the **Create case** button.

Example: Creating a Case

Here is an example of a Technical Support case (shown in two parts for readability). The lists that follow the form example explain some of your options and best practices.

Create Case Business support plan [Change](#)

Name

Account

CC ⓘ
Required for IAM users; use commas or semicolons to separate email addresses

Regarding* Account and Billing Support
 Service Limit Increase
 Technical Support

Service*

Category*
[How do I troubleshoot Instances with failed status checks?](#)
[How do I recover an unreachable Linux instance?](#)
[How do I troubleshoot instance errors?](#)
[Show more](#) ▾

Instance ID(s)

Severity* ⓘ

- **Contact Information.** In the **CC** box, enter the email addresses of people to be notified when the status of the case changes. If you are signed in as an IAM user, include your own email address. If you are signed in with your email address and password, you don't need to include your email address in the **CC** box.
- **Regarding.** Select the type of case you want to create. In this example, we select **Technical Support**.
- **Service.** If your question affects multiple services, choose the service that is most applicable. In this case, we select **Elastic Compute Cloud (EC2 - Linux)**.
- **Category.** Choose the most appropriate category. In this case, we're having trouble connecting to an instance, so we choose **Instance Issue**. When you select a category, links to information that might help to resolve your problem appear below the **Category** selection.

Based on your category choice, contextual text boxes often prompt for additional information. In this case, we're prompted to provide **Instance ID(s)**. In general, it's a good idea to provide resource IDs even when not prompted.

- **Severity.** All customers with a paid support plan can choose **General guidance** (1-day response time) or **System impaired** (12-hour response time) severity. Customers with a Business support plan can also choose **Production system impaired** (4-hour response) or **Production system down** (1-hour response). Enterprise plan customers can also choose **Business-critical system down** (15-minute response).

Note that response times are for first response from AWS Support, and do not apply to subsequent responses. For third-party issues, response times can be longer, depending on the availability of skilled personnel. For details, see [Choosing a Severity \(p. 4\)](#).

Subject*

Description Guidance
For troubleshooting ideas, see [Troubleshooting Instances](#).

If you are receiving an error, provide the detailed error message and a description of any changes that you may have made recently. Include the date, time, and time zone that you first observed the issue.

Description*

(limit 3, maximum 5 MB each)

failedStatus.png

Contact method* Web (we will get back to you within 12 hours)

Chat

Phone

- **Subject.** Treat this like the subject of an email message and sum up your issue as briefly as possible. In this case, we use the subject "Failed status checks."
- **Description.** This is the most important information that you provide to Support. For most service and category combinations, a prompt suggests information that is most helpful for the fastest resolution. For more guidance, see [Describing Your Problem \(p. 5\)](#).
- **Attachments.** Screen shots and other attachments (less than 5 MB each) are often helpful. In this case, we've added one to show the failed status check.
- **Contact method.** Select a contact method. The options vary depending on the type of case and your support plan. If you choose **Web**, you can read and respond to the case progress via Support Center. If you have a Business or Enterprise support plan, you can also select **Chat** or **Phone**. If you select **Phone**, you are prompted for a callback number and when you want Support to call you (when you submit the case, or in 5 minutes).
- **Submit.** Click the **Submit** button when your information is complete and you're ready to create the case.

Choosing a Severity

It might be tempting to open cases at the highest severity allowed by your support plan. However, we strongly encourage limiting the use of the highest severities to cases that cannot be worked around or that directly affect production applications. Plan ahead to avoid needing high-severity cases for general guidance questions. For information about how to build your services so that losing single resources does not affect your application, see [Building Fault-Tolerant Applications on AWS](#).

Here is a summary of severity levels, response times, and example problems. For more information about the scope of support for each AWS Support plan, see [AWS Support Features](#). **Note:** We make every reasonable effort to respond to your initial request within the indicated timeframe.

Severity	First-Response Time	Description / Support Plan
General guidance	24 hours	You have a general development question, or you want to request a feature. (Developer*, Business, and Enterprise support plans)
System impaired	12 hours	Non-critical functions of your application are behaving abnormally, or you have a time-sensitive development question. (Developer*, Business, and Enterprise support plans)
Production system impaired	4 hours	Important functions of your application are impaired or degraded. (Business and Enterprise support plans)
Production system down	1 hour	Your business is significantly impacted. Important functions of your application are unavailable. (Business and Enterprise support plans)
Business-critical system down	15 minutes	Your business is at risk. Critical functions of your application are unavailable. (Enterprise support plan)

* For the Developer plan, response targets are calculated in business hours. Business hours are generally defined as 8:00 AM to 6:00 PM in the customer country as set in the contact information of [My Account](#), excluding holidays and weekends. These times can vary in countries with multiple time zones.

Describing Your Problem

Your description should be as detailed as possible and include relevant resource information, along with anything else that would be beneficial, depending on the case. For example, to troubleshoot performance, include time stamps and logs. For feature requests or general guidance questions, include a description of your environment and purpose. In all cases, follow the **Description Guidance** that appears on your case submission form.

When you provide as much detail as possible, you increase the chances that your case will be resolved quickly.

Monitoring and Maintaining Your Case

You can monitor the status of your case in Support Center. A new case begins in the "Unassigned" state. When an engineer begins work on a case, the status changes to "Work in Progress." The engineer responds to your case, either to ask for more information ("Pending Customer Action") or to let you know that the case is being investigated ("Pending Amazon Action").

Whenever your case is updated, you receive email with the correspondence and a link to the case in Support Center—you cannot respond to case correspondence via email. When you are satisfied with the response or your problem has been solved, you can select **Close Case** in Support Center. If you do not respond within six days, the case is closed automatically. You can always reopen a resolved or closed case.

It is important to create a new case for a new issue or question. If case correspondence strays from the original question or issue, a support engineer might ask you to open a new case. When opening cases

related to old inquiries, it is helpful to include the related case number so that we can refer to previous correspondence.

Case History

Case history information is available for 12 months after creation.

Accessing AWS Support

There are two ways to access Support Center: by using the email address and password associated with your AWS account or—the preferred method—by using AWS Identity and Access Management (IAM). Customers with a Business or Enterprise support plan can access AWS Support and Trusted Advisor operations programmatically by using the [AWS Support API](#) (p. 10).

AWS Account

You can use your AWS account information to access Support Center. Simply sign in at <https://console.aws.amazon.com/support/home#/> and enter your email address and password. However, AWS recommends that you avoid using this method as much as possible and use IAM instead. For more information, see [Lock away your AWS account access keys](#).

IAM

You can use IAM to create individual users or groups, and then give them permission to perform actions and access resources in Support Center.

Note

IAM users who are granted Support access can see all cases that are created for the account.

By default, IAM users do not have access to Support Center. You give users access to your account's Support resources (Support Center cases and the AWS Support API) by attaching IAM policies to users, groups, or roles. For more information, see [IAM Users and Groups](#) and [Overview of AWS IAM Policies](#).

After you create IAM users, you can give those users individual passwords. They can then sign in to your account and work in Support Center by using an account-specific sign-in page. For more information, see [How IAM Users Sign In to Your AWS Account](#).

The easiest way to grant permission is to attach the AWS managed policy `AWSSupportAccess` to the group, role, or user. Support does not let you allow or deny access to individual actions; therefore, the `Action` element of a policy is always set to `support:*`. Similarly, Support does not provide resource-level access, so the `Resource` element is always set to `*`. An IAM user with Support permissions has access to all Support operations and resources.

For example, this policy statement grants access to Support:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "support:*",
      "Resource": "*"
    }
  ]
}
```

This policy statement denies access to Support:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "support:*",
      "Resource": "*"
    }
  ]
}
```

If the user or group already has a policy, you can add the Support-specific policy statement illustrated here to that policy.

Note

Access to Trusted Advisor in the AWS Management Console is controlled by a separate `trustedadvisor` IAM namespace. Access to Trusted Advisor via the AWS Support API is controlled by the `support` IAM namespace. For more information, see [Controlling Access to the Trusted Advisor Console](#).

AWS Trusted Advisor

[AWS Trusted Advisor](#) draws upon best practices learned from the aggregated operational history of serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment and makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to four Trusted Advisor checks. Customers with a Business or Enterprise support plan can view all Trusted Advisor checks. For more information, see [AWS Trusted Advisor](#).

For information about using Amazon CloudWatch Events to monitor the status of Trusted Advisor checks, see [Monitoring Trusted Advisor Check Results with Amazon CloudWatch Events \(p. 28\)](#).

Trusted Advisor can be accessed in the AWS Management Console; programmatic access to Trusted Advisor is available with the [AWS Support API \(p. 10\)](#).

Troubleshooting

For answers to common customer troubleshooting questions, see the [AWS Support Knowledge Center](#).

For troubleshooting Amazon CloudFront streaming video, you can use the [Amazon CloudFront Streaming Diagnostic Client](#).

For Windows, EC2 offers EC2Rescue, which allows customers to examine their Windows instances to help identify common problems, collect log files, and help Support troubleshoot issues. It can even be used to analyze boot volumes from non-functional instances. For more information, see [How can I use EC2Rescue to troubleshoot and fix common issues on my EC2 Windows instance?](#)

Service-specific Troubleshooting

Most AWS service documentation contains troubleshooting topics that can get you started before contacting Support. The following table provides links to troubleshooting topics in AWS documentation, arranged by service.

Service	Link
Amazon Web Services	Troubleshooting AWS Signature Version 4 Errors
Amazon AppStream	Troubleshoot Amazon AppStream
Auto Scaling	Troubleshooting Auto Scaling
AWS CloudFormation	Troubleshooting AWS CloudFormation
Amazon CloudFront	Troubleshooting Troubleshooting RTMP Distributions
AWS CloudHSM	Troubleshooting
Amazon CloudSearch	Troubleshooting Amazon CloudSearch
AWS CodeDeploy	Troubleshooting AWS CodeDeploy
AWS Data Pipeline	Troubleshooting
Amazon DevPay	Example Request to Use When Troubleshooting
AWS Direct Connect	Troubleshooting AWS Direct Connect
AWS Directory Service	Troubleshooting AWS Directory Service Administration Issues
Amazon DynamoDB	Troubleshooting
AWS Elastic Beanstalk	Troubleshooting
Amazon Elastic Compute Cloud (Amazon EC2)	Troubleshooting Instances Troubleshooting Windows Instances Troubleshooting VM Import/Export Troubleshooting API Request Errors Troubleshooting the AWS Management Pack Troubleshooting AWS Systems Manager for Microsoft SCVMM AWS Diagnostics for Microsoft Windows Server
Amazon EC2 Container Service (Amazon ECS)	Amazon ECS Troubleshooting
Elastic Load Balancing	Troubleshoot Your Application Load Balancers Troubleshoot Your Classic Load Balancer
Amazon EMR (Amazon EMR)	Troubleshoot a Cluster
Amazon ElastiCache	Troubleshooting Applications
AWS Flow Framework	Troubleshooting and Debugging Tips
AWS GovCloud (US)	Troubleshooting
AWS Identity and Access Management (IAM)	Troubleshooting IAM
Kinesis	Troubleshooting Amazon Kinesis Streams Producers Troubleshooting Amazon Kinesis Streams Consumers
AWS Lambda	Troubleshooting and Monitoring AWS Lambda Functions with CloudWatch
AWS OpsWorks	Debugging and Troubleshooting Guide

Service	Link
Amazon Redshift	Troubleshooting Queries Troubleshooting Data Loads Troubleshooting Connection Issues in Amazon Redshift Troubleshooting Amazon Redshift Audit Logging
Amazon Relational Database Service (Amazon RDS)	Troubleshooting Troubleshooting Applications
Amazon Silk	Troubleshooting
Amazon Simple Email Service (Amazon SES)	Troubleshooting Amazon SES
Amazon Simple Storage Service (Amazon S3)	Troubleshooting CORS Issues Handling REST and SOAP Errors
Amazon Simple Workflow Service (Amazon SWF)	AWS Flow Framework for Java: Troubleshooting and Debugging Tips AWS Flow Framework for Ruby: Troubleshooting and Debugging Workflows
AWS Storage Gateway	Troubleshooting Your Gateway
Amazon Virtual Private Cloud (Amazon VPC)	Troubleshooting
Amazon WorkMail	Troubleshooting the Amazon WorkMail Web Application
Amazon WorkSpaces	Troubleshooting Amazon WorkSpaces Administration Issues Troubleshooting Amazon WorkSpaces Client Issues
Amazon WorkSpaces Application Manager (Amazon WAM)	Troubleshooting Amazon WAM Application Issues

About the AWS Support API

The AWS Support API provides access to some of the features of the [AWS Support Center](#). AWS provides this access for [AWS Support](#) customers who have a Business or Enterprise support plan. The service currently provides two different groups of operations:

- [Support Case Management \(p. 10\)](#) operations to manage the entire life cycle of your AWS support cases, from creating a case to resolving it.
- [Trusted Advisor \(p. 11\)](#) operations to access the checks provided by [AWS Trusted Advisor](#).

For information about the operations and data types provided by AWS Support, see the [AWS Support API Reference](#).

Topics

- [Support Case Management \(p. 10\)](#)
- [Trusted Advisor \(p. 11\)](#)
- [Endpoint \(p. 11\)](#)
- [Support in AWS SDKs \(p. 11\)](#)

Support Case Management

Using the operations for support case management, you can perform these tasks:

- Open a support case.
- Get a list and detailed information about recent support cases.
- Narrow your search for support cases by dates and case identifiers, including cases that have been resolved.
- Add communications and file attachments to your cases, and add email recipients for case correspondence.
- Resolve your cases.

The AWS Support API supports CloudTrail logging for support case management operations. For more information, see [Logging AWS Support API Calls with AWS CloudTrail \(p. 25\)](#).

For example Java code that demonstrates how to manage the entire life cycle of an AWS Support case, see [Programming the Life of an AWS Support Case \(p. 12\)](#).

Trusted Advisor

Using the Trusted Advisor operations, you can perform these tasks:

- Get names and identifiers for the checks that Trusted Advisor offers.
- Request that a Trusted Advisor check be run against your account and resources.
- Obtain summaries and detailed information for your Trusted Advisor checks.
- Request that Trusted Advisor checks be refreshed.
- Obtain the status of each Trusted Advisor check you have requested.

The AWS Support API supports CloudWatch Events for Trusted Advisor operations. For more information, see [Monitoring Trusted Advisor Check Results with Amazon CloudWatch Events \(p. 28\)](#).

For an example that uses the Trusted Advisor operations, see [Using Trusted Advisor as a Web Service \(p. 21\)](#).

Endpoint

Use this endpoint to access AWS Support:

- <https://support.us-east-1.amazonaws.com>

Warning

The AWS Support endpoint creates cases in the production database. Include a subject line such as "TEST CASE--Please ignore" when you call [CreateCase](#) for testing, and close the test cases you create by calling [ResolveCase](#).

For additional information about using AWS endpoints, see [Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

Support in AWS SDKs

The AWS Command Line Interface, the AWS Tools for Windows PowerShell, and the AWS Software Development Kits (SDKs) include support for the AWS Support API:

- [AWS CLI](#)
- [AWS Tools for Windows PowerShell](#)
- [AWS SDK for Java](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for .NET](#)
- [AWS SDK for PHP](#)
- [AWS SDK for Python \(Boto\)](#)
- [AWS SDK for Ruby](#)

Programming the Life of an AWS Support Case

The AWS Support API enables you to create cases and add correspondence to them throughout investigations of your issues and interactions with AWS Support staff. This topic demonstrates the use of operations in the AWS Support service, which models much of the behavior of the [AWS Support Center](#).

For detailed information, see the [AWS Support API Reference](#).

Topics

- [Overview \(p. 12\)](#)
- [Create an AWS Support Client \(p. 13\)](#)
- [Discover AWS Services and Issue Severity Levels \(p. 13\)](#)
- [Create an Attachment Set \(p. 14\)](#)
- [Create a Support Case \(p. 15\)](#)
- [Retrieve and Update Support Case Communications \(p. 17\)](#)
- [Retrieve All Support Case Information \(p. 19\)](#)
- [Resolve a Support Case \(p. 20\)](#)

Overview

This topic uses Java code examples to demonstrate the use of AWS Support. For more information about SDK support, see [Sample Code & Libraries](#).

Note

If you encounter service limits with your calls to AWS Support, follow the recommendations in [Error Retries and Exponential Backoff in AWS](#)

Using IAM with the AWS Support API

AWS Identity and Access Management (IAM) is supported by the AWS Support API. For more information, see [Accessing AWS Support \(p. 6\)](#).

Create an AWS Support Client

The following Java code snippet shows how to create an `AWSSupportClient`, which is used to call the `AWSSupportService`. The `createClient` method gets AWS credentials by calling the `AWSSupportClient()` constructor with no parameters, which retrieves credentials from the credentials provider chain. For more information on this process, see [Tutorial: Grant Access Using an IAM Role and the AWS SDK for Java](#) in the *AWS SDK for Java*.

For more information on AWS credentials, see [AWS Security Credentials](#) in the *AWS General Reference*.

```
private static AWSSupportClient createClient()
{
    AWSSupportClient client = new AWSSupportClient();
    client.setEndpoint("https://support.us-east-1.amazonaws.com");
    return client;
}
```

Discover AWS Services and Issue Severity Levels

The AWS Support Java client provides a `CreateCaseRequest` type to submit a case programmatically to AWS Support. The `CreateCaseRequest` structure is populated with the request parameters and passed to the `createClient` method on the `AWSSupportClient` instance. These parameters include codes that specify the AWS service and case severity.

The following Java code snippet demonstrates calls to the AWS Support [DescribeServices](#) and [DescribeSeverityLevel](#) actions:

```
// DescribeServices example

public static void getServiceCodes(AWSSupportClient client)
{
    DescribeServicesResult result = client.describeServices();
    for (Service service : result.getServices())
    {
        System.out.println("Service code (name): " +
            service.getCode() + service.getName() + "");
        for (Category category : service.getCategories())
        {
            System.out.println("    Category code (name): " +
                category.getCode() + "(" + category.getName() + ")");
        }
    }
}

// DescribeSeverityLevels example

public static void getSeverityLevels(AWSSupportClient client)
{
    DescribeSeverityLevelsResult result = client.describeSeverityLevels();
    for (SeverityLevel level : result.getSeverityLevelsList())
    {
        System.out.println("Severity level (name): " +
            level.getCode() + level.getName() + "");
    }
}
```

Each call returns a list of JSON-formatted objects. `DescribeServices` returns service codes and their corresponding names, and `DescribeSeverityLevels` returns severity levels and their corresponding

names. In addition, `DescribeServices` also returns a list of AWS Support categories that apply to each AWS service. These categories are also used to open a support case by using `createCase`. Although these values can also be obtained from the AWS Support site itself, the AWS Support service always returns the most recent version of this information.

Create an Attachment Set

If you want to attach files to the case, you must add the attachments to an attachment set before creating the case. You can add up to 3 attachments to a set, and the maximum size of any attachment in the set is 5 MB. For more information, see [AddAttachmentsToSet](#).

The following Java code snippet creates a text file attachment, adds it to an attachment set, and gets the ID of the attachment set for adding to the case.

```
public static String createAttachmentSet() throws IOException
{
    BufferedReader reader =
        new BufferedReader(new InputStreamReader(System.in));

    // Get content and file name for an attachment.
    System.out.println("Enter text content for an attachment to the case: ");
    String attachmentcontent = null;
    try
    {
        attachmentcontent = reader.readLine().trim();
    }
    catch (IOException e)
    {
        e.printStackTrace();
        System.exit(1);
    }

    System.out.println("Enter the file name for the attachment: ");
    String attachmentfilename = null;
    try
    {
        attachmentfilename = reader.readLine().trim();
    }
    catch (IOException e)
    {
        e.printStackTrace();
        System.exit(1);
    }

    // Create the attachment.
    Attachment attachment1 = new Attachment();
    attachment1.setData(ByteBuffer.wrap(attachmentcontent.getBytes()));
    attachment1.setFileName("attachmentfilename");

    // Add the attachment to an array list.
    List<Attachment> attachments = new ArrayList<Attachment>();
    attachments.add(attachment1);

    // Create an attachment set and add the attachment array list to it.
    AddAttachmentsToSetRequest addAttachmentsToSetRequest =
        new AddAttachmentsToSetRequest();
    addAttachmentsToSetRequest.setAttachments(attachments);

    AddAttachmentsToSetResult addAttachmentsToSetResult =
        client.addAttachmentsToSet(addAttachmentsToSetRequest);
}
```

```
// Get the ID of the attachment set.  
String attachmentsetid = addAttachmentsToSetResult.getAttachmentSetId();  
System.out.println("Attachment ID: " + attachmentsetid);  
return attachmentsetid;  
}
```

Create a Support Case

To create an AWS Support case using the AWS Support service, populate a `CreateCaseRequest` instance with the following information:

- `ServiceCode`. The AWS Support service code you obtained by calling `DescribeServices` as described in the previous section.
- `CategoryCode`. The category code that describes the type of issue the support case concerns.
- `Language`. A code for the language in which AWS Support provides support. Currently, AWS supports English (`en`) and Japanese (`ja`).
- `CcEmailAddresses`. A list of email addresses to receive copies of subsequent communications.
- `CommunicationBody`. Text for the body of the initial case submission.
- `Subject`. A title for the support case.
- `SeverityCode`. One of the values returned by the call to `DescribeSeverityLevels`.
- `AttachmentSetId`. (Optional) The ID of a set of file attachments to include with the case. The `AddAttachmentsToSet` operation returns the ID.

The following Java code snippet collects values for each of the case creation parameters from the command line. It then populates a `CreateCaseRequest` instance and passes them to AWS Support by calling the `createCase` method on an `AWSSupportClient` instance. If the call is successful, it returns an AWS Support `CaseId` value in the format:

- `case-123456789012-muen-2012-74a757cd8cf7558a`

Note

AWS Support provides both `CaseId` and `DisplayId` fields. The `DisplayId` field corresponds to the case number that is displayed on the AWS Support site. The `CaseId` field is for use in programmatic interactions with the AWS Support service. Both fields are exposed on the `CaseDetails` data type.

```
public static void createCase(AWSSupportClient client) throws IOException  
{  
    BufferedReader reader =  
        new BufferedReader(new InputStreamReader(System.in));  
  
    System.out.println("Enter an AWS Service code: ");  
    String servicecode = null;  
    try  
    {  
        servicecode = reader.readLine().trim();  
    }  
    catch (IOException e)  
    {  
        e.printStackTrace();  
        System.exit(1);  
    }  
}
```

```
System.out.println("Enter a category code: ");
String categorycode = null;
try
{
    categorycode = reader.readLine().trim();
}
catch (IOException e)
{
    e.printStackTrace();
    System.exit(1);
}

System.out.println("Enter a language code, 'en' for English: ");
String language = null;
try
{
    language = reader.readLine().trim();
}
catch (IOException e)
{
    e.printStackTrace();
    System.exit(1);
}

System.out.println("Enter an email address to copy on correspondence: ");
String ccemailaddress = null;
try
{
    ccemailaddress = reader.readLine().trim();
}
catch (IOException e)
{
    e.printStackTrace();
    System.exit(1);
}

System.out.println("Enter body text for the case: ");
String communicationbody = null;
try
{
    communicationbody = reader.readLine().trim();
}
catch (IOException e)
{
    e.printStackTrace();
    System.exit(1);
}

System.out.println("Enter a subject for the case: ");
String casesubject = null;
try
{
    casesubject = reader.readLine().trim();
}
catch (IOException e)
{
    e.printStackTrace();
    System.exit(1);
}

System.out.println("Enter the severity code for the case: ");
String severitycode = null;
try
{
    severitycode = reader.readLine().trim();
}
}
```

```
catch (IOException e)
{
    e.printStackTrace();
    System.exit(1);
}

System.out.println("Enter the attachment set ID for the case: ");
String attachmentsetid = null;
try
{
    attachmentsetid = reader.readLine().trim();
}
catch (IOException e)
{
    e.printStackTrace();
    System.exit(1);
}

CreateCaseRequest request = new CreateCaseRequest()
    .withServiceCode(servicecode)
    .withCategoryCode(categorycode)
    .withLanguage(language)
    .withCcEmailAddresses(ccemailaddress)
    .withCommunicationBody(communicationbody)
    .withSubject(casesubject)
    .withSeverityCode(severitycode)
    .withAttachmentSetId(attachmentsetid);

CreateCaseResult result = client.createCase(request);
System.out.println("CreateCase() Example: Case created with ID "
    + result.getCaseId());
}
```

Retrieve and Update Support Case Communications

AWS Support cases almost always result in communication between the customer and AWS Support professionals. AWS Support provides the [DescribeCommunications](#) and [DescribeAttachment](#) operations to retrieve this correspondence, and the [AddAttachmentsToSet](#) and [AddCommunicationToCase](#) operations to update the case. These operations use the [Communication](#) data type to pass updates to the service and return them to your code.

The following Java code snippet adds communication to an AWS Support case. In the example, a private `PrintCommunications` method is provided for your convenience.

```
public static void addCommunication(AWSSupportClient client)
{
    System.out.println("Enter the CaseID for the case you want to update.");
    BufferedReader reader =
        new BufferedReader(new InputStreamReader(System.in));
    String caseid = null;
    try
    {
        caseid = reader.readLine().trim();
    }
    catch (IOException e)
    {
        e.printStackTrace();
        System.exit(1);
    }
}
```

```
    }

    System.out.println("Enter text you want to add to this case.");
    String addcomm = null;
    try
    {
        addcomm = reader.readLine().trim();
    }
    catch (IOException e)
    {
        e.printStackTrace();
        System.exit(1);
    }

    AddCommunicationToCaseRequest request =
        new AddCommunicationToCaseRequest().withCaseId(caseid)
                                           .withCommunicationBody(addcomm);
    client.addCommunicationToCase(request);

    System.out.println(
        "AddCommunication() Example: Call GetCommunications() " +
        "if you want to see if the communication was added.");
}

// DescribeCommunications example

public static void getCommunications(AWSSupportClient client)
    throws IOException
{
    BufferedReader reader =
        new BufferedReader(new InputStreamReader(System.in));
    String caseNumber = null;

    System.out.println("Enter an AWS CaseID");
    caseNumber = reader.readLine().trim();

    {
        DescribeCommunicationsRequest request =
            new DescribeCommunicationsRequest()
                .withCaseId(caseNumber.toString());

        DescribeCommunicationsResult result =
            client.describeCommunications(request);
        printCommunications(result.getCommunications());

        // Get more pages.
        while (result.getNextToken() != null)
        {
            request.setNextToken(result.getNextToken());
            result = client.describeCommunications(request);
            printCommunications(result.getCommunications());
            System.out.println(
                "GetCommunications() Example: Case communications retrieved"
                + " for case number " + request.getCaseId().toString());
        }
    }
}

private static void printCommunications(List<Communication> communications)
{
    for (Communication communication : communications)
    {
        System.out.println("SubmittedBy: " + communication.getSubmittedBy());
        System.out.println("  Body: " + communication.getBody());
    }
}
```

```
}
```

Note

`DescribeCommunications` returns the five most recent communications from a support case. Also, `DescribeCommunications` takes a list of `CaseId` values, enabling you to retrieve communications for multiple cases in a single call.

Retrieve All Support Case Information

You can retrieve all information associated with your AWS Support cases by calling the `DescribeCases` action. You populate a `DescribeCasesRequest` data type with a list of `ClientId` values, which are returned by each case when a successful `createCase` request returns.

The following Java code snippet accepts `CaseId` values from the console and populates a `DescribeCasesRequest` instance for use by the `DescribeCases` action. A private `printCases` method is provided for your convenience.

```
public static void getCases(AWSSupportClient client)
{
    BufferedReader reader =
        new BufferedReader(new InputStreamReader(System.in));

    System.out.println("Enter an AWS Support Case ID");
    String caseid = null;
    try
    {
        caseid = reader.readLine().trim();
    }
    catch (IOException e)
    {
        e.printStackTrace();
        System.exit(1);
    }

    DescribeCasesRequest request = new DescribeCasesRequest();
    request.withCaseIdList(caseid);

    DescribeCasesResult result = client.describeCases(request);
    printCases(result.getCases());

    // Get more pages.
    while (result.getNextToken() != null)
    {
        request.setNextToken(result.getNextToken());
        result = client.describeCases(request);
        printCases(result.getCases());
    }
}

private static void printCases(List<CaseDetails> caseDetailsList)
{
    for (CaseDetails caseDetails : caseDetailsList)
    {
        System.out.println(
            "Case ID: " + caseDetails.getCaseId()); // This ID is for API use.
        System.out.println(
            "  Display ID: " + caseDetails.getDisplayId());
        // This ID is displayed on the AWS Support website.
        System.out.println("  Language: " + caseDetails.getLanguage());
    }
}
```

```
        System.out.println(" Status: " + caseDetails.getStatus());
        System.out.println(" Subject: " + caseDetails.getSubject());
        System.out.println("Recent Communications: " +
            caseDetails.getRecentCommunications());
    }
}
```

Note

The `DescribeCases` operation takes parameters that allow you to control the number of cases, types of cases, and amount of detail that is retrieved. For more information, see [DescribeCases](#).

Resolve a Support Case

AWS Support provides a [ResolveCase](#) action to resolve your own support cases. The following Java code example demonstrates its use.

```
public static void resolveSupportCase(AWSSupportClient client)
{
    System.out.println(
        "Enter the AWS Support case ID for the case you want to resolve.");
    BufferedReader BR = new BufferedReader(new InputStreamReader(System.in));

    String caseid = null;
    try
    {
        caseid = BR.readLine().trim();
    }
    catch (IOException e)
    {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    ResolveCaseResult rcr =
        client.resolveCase(new ResolveCaseRequest().withCaseId(caseid));
    System.out.println("Initial case status: " + rcr.getInitialCaseStatus());
    System.out.println("Final case status: " + rcr.getFinalCaseStatus());
}
```

Using Trusted Advisor as a Web Service

The AWS Support service enables you to write applications that interact with [AWS Trusted Advisor](#). This topic shows you how to get a list of Trusted Advisor checks, refresh one of them, and obtain the detailed results from the check. These tasks are demonstrated in Java. For information about support for other languages, see [Tools for Amazon Web Services](#).

Topics

- [Get the List of Available Trusted Advisor Checks \(p. 21\)](#)
- [Request a Trusted Advisor Check Result \(p. 22\)](#)
- [Poll a Trusted Advisor Check for Status Changes \(p. 22\)](#)
- [Print Details of a Trusted Advisor Check \(p. 23\)](#)

Get the List of Available Trusted Advisor Checks

The following Java code snippet creates an instance of an AWS Support client that you can use to call all Trusted Advisor actions. Next, the code gets the list of Trusted Advisor checks and their corresponding `CheckId` values by calling the [DescribeTrustedAdvisorChecks](#) action. You can use this information to build user interfaces that enable users to select the check they want to run or refresh.

```
private static AWSSupportClient createClient()
{
    AWSSupportClient client = new AWSSupportClient();
    client.setEndpoint("https://support.us-east-1.amazonaws.com");
    return client;
}

public static void getTaChecks(AWSSupportClient client)
{
    DescribeTrustedAdvisorChecksResult result =
        client.describeTrustedAdvisorChecks(
            new DescribeTrustedAdvisorChecksRequest()
                .withLanguage("en"));
}
```



```
for (TrustedAdvisorCheckDescription checkDescription :
    result.getChecks())
{
    System.out.println(checkDescription.getId());
    System.out.println(checkDescription.getName());
}
```

Request a Trusted Advisor Check Result

After you have selected the check you want to run, you submit a request by using the [DescribeTrustedAdvisorCheckResult](#) action.

The following Java code snippet uses the `DescribeTrustedAdvisorChecksResult` instance referenced by the variable `result`, which was obtained in the preceding code snippet. Rather than defining a check interactively through a user interface, the snippet simply submits a request for the first check in the list to be run by specifying an index value of 0 in each `result.getChecks().get(0)` call. Next, the code defines an instance of `DescribeTrustedAdvisorCheckResultRequest`, which it passes to an instance of `DescribeTrustedAdvisorCheckResultResult` called `checkResult`. You can use the member structures of this data type to view the results of the check.

```
final String checkId = result.getChecks().get(0).getId();
final String checkName = result.getChecks().get(0).getName();

System.out.println("The following check is being refreshed: " +
    checkId + ":" + " " + checkName);

DescribeTrustedAdvisorCheckResultRequest checkResultRequest =
    new DescribeTrustedAdvisorCheckResultRequest().withCheckId(checkId);
DescribeTrustedAdvisorCheckResultResult checkResult =
    client.describeTrustedAdvisorCheckResult(checkResultRequest);
```

Poll a Trusted Advisor Check for Status Changes

After you have submitted the request to run a Trusted Advisor check, you use the [DescribeTrustedAdvisorCheckRefreshStatuses](#) action to request the status of a check that you've started.

The following Java code snippet gets the status of the check requested in the following section, using the value corresponding in the `checkId` variable. In addition, the code demonstrates several other uses of the Trusted Advisor service:

1. You can call `getMillisUntilNextRefreshable` by traversing the objects contained in the `DescribeTrustedAdvisorCheckRefreshStatusesResult` instance. You can use the value returned to test whether you want your code to proceed with refreshing the check.
2. If `timeUntilRefreshable` equals zero, you can request a refresh of the check.
3. Using the status returned, you can continue to poll for status changes; the code snippet sets the polling interval to a recommended ten seconds. If the status is either `enqueued` or `in_progress`, the loop returns and requests another status. If the call returns `successful`, the loop terminates.
4. Last, the code returns an instance of a `DescribeTrustedAdvisorCheckResultResult` data type that you can use to traverse the information produced by the check.

```
String status = "";

do
{
    DescribeTrustedAdvisorCheckRefreshStatusesResult
        describeStatusResult =
            client.describeTrustedAdvisorCheckRefreshStatuses(
                new DescribeTrustedAdvisorCheckRefreshStatusesRequest()
                    .withCheckIds(java.util.Arrays.asList(checkId)));
    final long timeUntilRefreshable =
        describeStatusResult.getStatuses().get(0)
            .getMillisUntilNextRefreshable();

    if (timeUntilRefreshable != 0)
    {
        System.out.println(
            "Can't refresh; check was recently refreshed. "
            + "Time until refreshable: "
            + timeUntilRefreshable);
    }

    DescribeTrustedAdvisorCheckRefreshStatusesResult
        describeRefreshStatusResult =
            client.describeTrustedAdvisorCheckRefreshStatuses(
                new DescribeTrustedAdvisorCheckRefreshStatusesRequest()
                    .withCheckIds(java.util.Arrays.asList(checkId)));

    status = describeRefreshStatusResult.getStatuses().get(0).getStatus();

    System.out.println("Check status: " + status.toString());

    try
    {
        Thread.sleep(10000);
    }
    catch (InterruptedException e)
    {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    // Fetch the updated result.
    DescribeTrustedAdvisorCheckResultResult result1 =
        client.describeTrustedAdvisorCheckResult(
            new DescribeTrustedAdvisorCheckResultRequest()
                .withCheckId(checkId));

    return result1;
}
while (status.equals("enqueued") || status.equals("in_progress"));
```

Print Details of a Trusted Advisor Check

The following Java code snippet iterates over the `DescribeTrustedAdvisorCheckResultResult` instance returned in the previous section to get a list of resources flagged by the Trusted Advisor check.

```
// Print ResourceIds for flagged resources.
for (TrustedAdvisorResourceDetail flaggedResource :
    result1.getResult().getFlaggedResources())
{
```

```
System.out.println(  
    "The resource for this ResourceID has been flagged: " +  
    flaggedResource.getResourceId());  
}
```

Logging AWS Support API Calls with AWS CloudTrail

AWS Support is integrated with CloudTrail, a service that captures specific API calls and delivers log files to an Amazon S3 bucket that you specify. CloudTrail captures API calls made from the AWS Support console or from your code to the AWS Support APIs. With the information collected by CloudTrail, you can determine the request that was made to AWS Support, the IP address from which the request was made, who made the request, when it was made, and so on.

To learn more about CloudTrail, including how to configure and enable it, see the [AWS CloudTrail User Guide](#).

AWS Support Information in CloudTrail History

The CloudTrail API activity history feature lets you look up and filter events captured by CloudTrail. You can look up events related to the creation, modification, or resolution of support cases in your AWS account. Events can be looked up by using the CloudTrail console, or programmatically by using the AWS SDKs or AWS CLI ([lookup-events](#)).

The following actions are supported:

- [AddAttachmentsToSet](#)
- [AddCommunicationToCase](#)
- [CreateCase](#)
- [ResolveCase](#)

AWS Support Information in CloudTrail Logging

When CloudTrail logging is enabled in your AWS account, API calls made to specific AWS Support actions are tracked in CloudTrail log files. AWS Support actions are written with other AWS service records. CloudTrail determines when to create and write to a new file based on a time period and file size.

The following actions are supported:

- [AddAttachmentsToSet](#)

- [AddCommunicationToCase](#)
- [CreateCase](#)
- [DescribeAttachment](#)
- [DescribeCases](#)
- [DescribeCommunications](#)
- [DescribeServices](#)
- [DescribeSeverityLevels](#)
- [ResolveCase](#)

Every log entry contains information about who generated the request. The user identity information in the log entry helps you determine the following:

- Whether the request was made with root or IAM user credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see the [CloudTrail userIdentity Element](#).

You can store your log files in your Amazon S3 bucket for as long as you want, but you can also define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, your log files are encrypted with Amazon S3 server-side encryption (SSE).

If you want to be notified upon log file delivery, you can configure CloudTrail to publish Amazon SNS notifications when new log files are delivered. For more information, see [Configuring Amazon SNS Notifications for CloudTrail](#).

You can also aggregate AWS Support log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket.

For more information, see [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#).

Understanding AWS Support Log File Entries

CloudTrail log files contain one or more log entries. Each entry lists multiple JSON-formatted events. A log entry represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. The log entries are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the [CreateCase](#) action.

```
{
  "Records": [
    {
      "eventVersion": "1.04",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/janedoe",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "janedoe",
        "sessionContext": {
```

```
        "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2016-04-13T17:51:37Z"
        },
        "invokedBy": "signin.amazonaws.com"
    },
    "eventTime": "2016-04-13T18:05:53Z",
    "eventSource": "support.amazonaws.com",
    "eventName": "CreateCase",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "198.51.100.15",
    "userAgent": "signin.amazonaws.com",
    "requestParameters": {
        "severityCode": "low",
        "categoryCode": "other",
        "language": "en",
        "serviceCode": "support-api",
        "issueType": "technical"
    },
    "responseElements": {
        "caseId": "case-111122223333-muen-2016-c3f2077e504940f2"
    },
    "requestID": "58c257ef-01a2-11e6-be2a-01c031063738",
    "eventID": "5aa34bfc-ad5b-4fb1-8a55-2277c86e746a",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
],
...
}
```

Monitoring Trusted Advisor Check Results with Amazon CloudWatch Events

You can use Amazon CloudWatch Events to detect and react to changes in the status of Trusted Advisor checks. Then, based on rules you create, CloudWatch Events invokes one or more target actions when a check status changes to the value you specify in a rule. Depending on the type of status change, you might want to send notifications, capture status information, take corrective action, initiate events, or take other actions. You can select the following types of targets when using CloudWatch Events as a part of your Trusted Advisor workflow:

- AWS Lambda functions
- Kinesis streams
- Amazon SQS queues
- Built-in targets (CloudWatch alarm actions)
- Amazon SNS topics

The following are some use cases:

- Use a Lambda function to pass a notification to a Slack channel when check status changes.
- Push data about checks to an Kinesis stream to support comprehensive, real-time status monitoring.

For examples of using CloudWatch Events and Lambda functions to automate the response to Trusted Advisor check results, see [Trusted Advisor Tools](#).

The remainder of this topic describes the basic procedure for creating a CloudWatch Events rule for Trusted Advisor. Before you create event rules for Trusted Advisor, however, you should do the following:

- Familiarize yourself with events, rules, and targets in CloudWatch Events. For more information, see [What Is Amazon CloudWatch Events?](#) and [New CloudWatch Events – Track and Respond to Changes to Your AWS Resources](#).
- Create the target or targets you will use in your event rules.

To create a CloudWatch Events rule for Trusted Advisor:

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation bar, choose the **US East (N. Virginia)** Region.
3. In the navigation pane, choose **Events**.
4. Choose **Create rule**, and then under **Event Source**, for **Service Name**, choose **Trusted Advisor**.
5. Specify status values:
 - To make a rule that applies to all status values, choose **Check Item Refresh Status**, and then choose **Any status** (the default).
 - To make a rule that applies to some status values only, choose **Specific status(es)**, and then choose one or more status values from the list.
6. Specify Trusted Advisor checks:
 - To make a rule that applies to all Trusted Advisor checks, choose **Any check**.
 - To make a rule that applies to some checks only, choose **Specific check(s)**, and then choose one or more check names from the list.
7. Specify AWS resources:
 - To make a rule that applies to all resources, choose **Any resource ID**.
 - To make a rule that applies to one or more resources only, choose **Specific resource ID(s) by ARN**, and then type the ARNs of the resources.
8. Review your rule setup to make sure it meets your event-monitoring requirements.
9. In the **Targets** area, choose **Add target***.
10. In the **Select target type** list, choose the type of target you have prepared to use with this rule, and then configure any additional options required by that type.
11. Choose **Configure details**.
12. On the **Configure rule details** page, type a name and description for the rule. To enable the rule as soon as it is created, choose the **State** box.
13. If you're satisfied with the rule, choose **Create rule**.

Document History

The following table describes the important changes to the documentation since the last release of the AWS Support service.

- **API version:** 2013-04-15
- **Latest documentation update:** November 18, 2016

Change	Description	Date Changed
Monitoring Trusted Advisor Check Results with CloudWatch Events	Added new section.	November 18, 2016
Case Management	Updated the names of case severity levels.	October 27, 2016
Logging AWS Support API Calls with AWS CloudTrail	Added new section.	April 21, 2016
Getting Started: Troubleshooting	Added more troubleshooting links.	May 19, 2015
Getting Started: Troubleshooting	Added more troubleshooting links.	November 18, 2014
Getting Started: Case Management	Updated to reflect Support Center in the AWS Management Console.	October 30, 2014
Programming the Life of an AWS Support Case	Added information about new API elements for adding attachments to cases and for omitting case communications when retrieving case history.	July 16, 2014
Accessing AWS Support	Removed named support contacts as an access method.	May 28, 2014
Getting Started	Added the Getting Started section.	December 13, 2013
Initial publication	New AWS Support service released.	April 30, 2013

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.