AWS Support

User Guide

API Version 2013-04-15

AWS Support: User Guide

Table of Contents

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. For technical support and more resources to plan, deploy, and improve your AWS environment, you can select a support plan that best aligns with your AWS use case.

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
- AWS Identity and Access Management (p. 7) (IAM) for controlling individuals' access to AWS Support

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 Trusted Advisor (p. 8), which inspects customer environments. Then, Trusted Advisor identifies opportunities to save money, close security gaps, and improve system reliability and performance.
- An API for interacting with Support Center and Trusted Advisor. This API allows 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. Also, help with the 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 after analysis, provide architectural and scaling guidance for an event.
- 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, aren't 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 linked to your AWS account. To log in with other credentials, see Accessing AWS Support (p. 6).

There are three types of cases you can open:

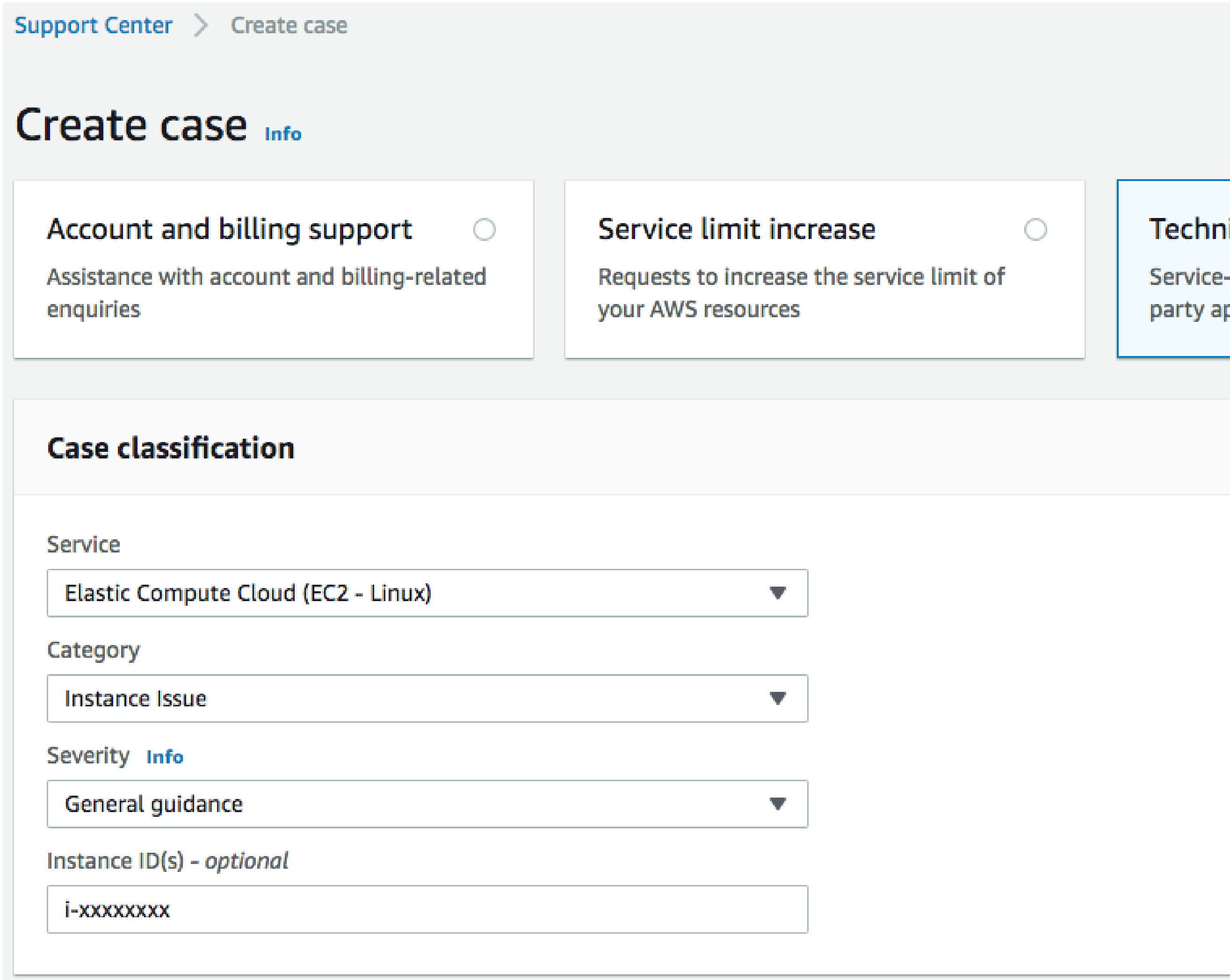
- **Account and Billing Support** cases are available to all AWS customers. This case type connects you to customer service for help with billing and account-related questions.
- **Service Limit Increase** requests are also available to all AWS customers. For information on the default service limits, 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 using 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, choose the **Create case** button.

Example: Creating a Case

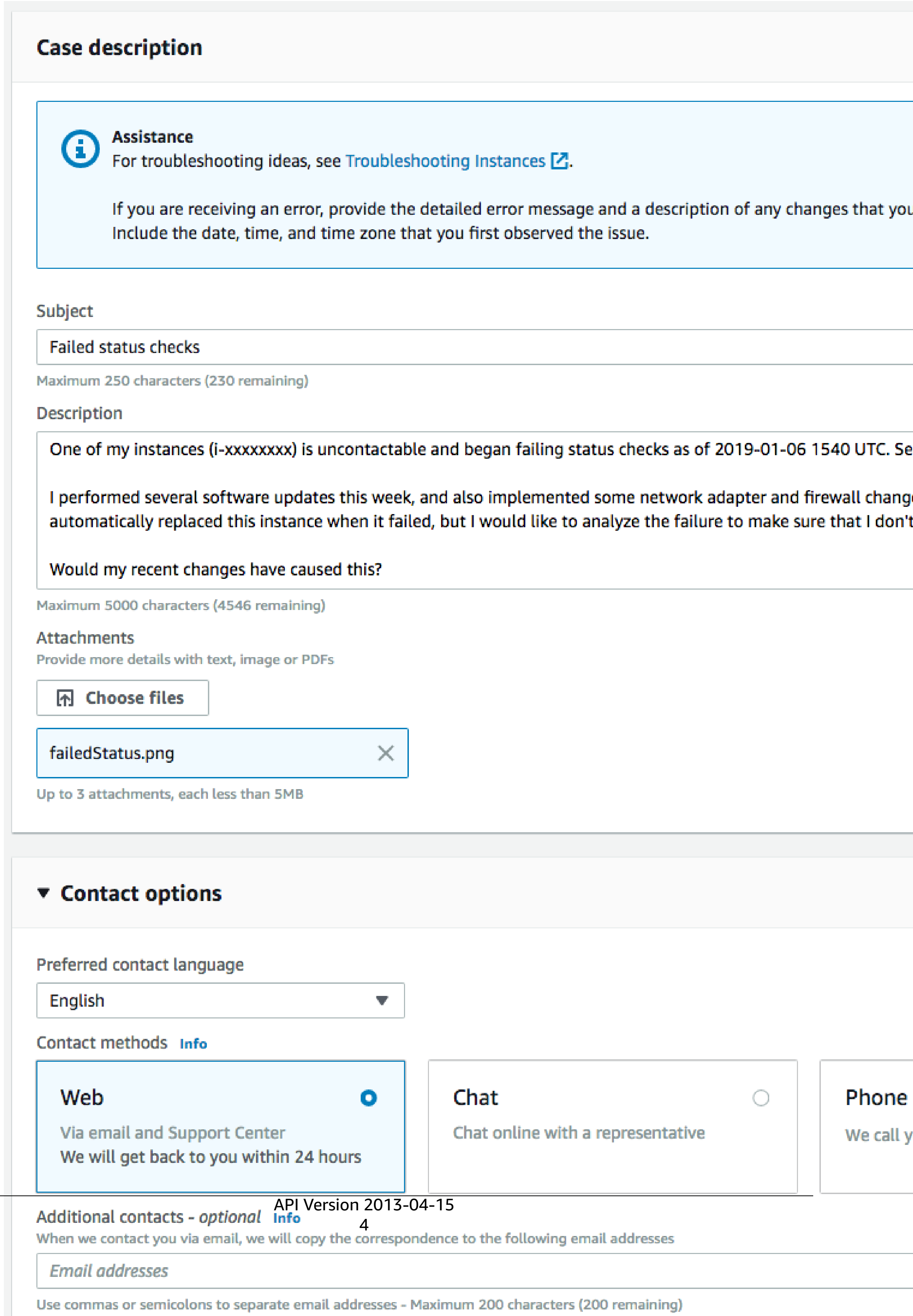
Here is an example of a Technical Support case (shown in two parts for readability). The lists that follow the form example explain some of your options and best practices.



- **Service**. If your question affects multiple services, choose the service that's most applicable. In this case, select **Elastic Compute Cloud (EC2 - Linux)**.
- **Category**. Choose the category that best fits your use case. In this case, there's trouble connecting to an instance, so choose **Instance Issue**. When you select a category, links to information that might help to resolve your problem appear below the **Category** selection.
- **Severity**. Customers with a paid support plan can choose the **General guidance** (1-day response time) or **System impaired** (12-hour response time) severity level. Customers with a Business support plan can also choose **Production system impaired** (4-hour response) or **Production system down** (1-hour response). And customers with an Enterprise plan can choose **Business-critical system down** (15-minute response).

  Response times are for first response from AWS Support. These response times don't 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. 5).

  Note
  Based on your category choice, you might be prompted for additional information. In this case, you're prompted to provide the **Instance IDs**. In general, it's a good idea to provide resource IDs, even when not prompted.


Case description
Assistance
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
Include the date, time, and time zone that you first observed the issue.
Subject
Failed status checks
Maximum 250 characters (230 remaining)
Description
One of my instances (i-xxxxxxxx) is uncontactable and began failing status checks as of 2019-01-06 1540 UTC. Se
I performed several software updates this week, and also implemented some network adapter and firewall chang
automatically replaced this instance when it failed, but I would like to analyze the failure to make sure that I don't
Would my recent changes have caused this?
Maximum 5000 characters (4546 remaining)
Attachments
Provide more details with text, image or PDFs
Choose files
failedStatus.png
Up to 3 attachments, each less than 5MB
Contact options
Preferred contact language
English
Contact methods Info
Web
Via email and Support Center
We will get back to you within 24 hours
Chat
Chat online with a representative
Phone
We call y
Additional contacts - optional Info
When we contact you via email, we will copy the correspondence to the following email addresses
Email addresses
Use commas or semicolons to separate email addresses - Maximum 200 characters (200 remaining)

- **Subject**. Treat this like the subject of an email message—briefly describe your issue. In this case, use the subject `Failed status checks`.
- **Description**. This is the most important information that you provide to AWS Support. For most service and category combinations, a prompt suggests information that's most helpful for the fastest resolution. For more guidance, see Describing Your Problem (p. 6).
- **Attachments**. Screen shots and other attachments (less than 5 MB each) can be helpful. In this case, an image is added that shows the failed status check.
- **Contact methods**. 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 in Support Center. If you have a Business or Enterprise support plan, you can also select **Chat** or **Phone**. If you select **Phone**, you're prompted for a callback number.
- **Additional contacts**. Provide the email addresses of people to be notified when the status of the case changes. If you're signed in as an IAM user, include your own email address. If you're signed in with your email address and password, you don't need to include your email address in this box.

  Note
  If you have the Basic support plan, the **Additional contacts** box isn't available. However, the **Operational** contact specified in the **Alternate Contacts** section of the My Account page receives copies of the case correspondence.
- **Case Type**. Select the type of case you want to create from the three boxes at the top of the page. In this example, select **Technical Support**.

  Note
  If you have the Basic support plan, you can't create a technical support case.
- **Submit**. Choose **Submit** when your information is complete. Choosing **Submit** creates the case.

Choosing a Severity

You might want to always open cases at the highest severity allowed by your support plan. However, we strongly encourage that you limit the use of the highest severities to cases that can't be worked around or that directly affect production applications. Plan ahead to avoid high-severity cases for general guidance questions. For information about building your services so that losing single resources doesn't 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)

Severity	First-Response Time	Description / Support Plan
Production system down	1 hour	Your business is significantly impacted. Important functions of your application aren't available. (Business and Enterprise support plans)
Business-critical system down	15 minutes	Your business is at risk. Critical functions of your application aren't available. (Enterprise support plan)

* For the Developer plan, response targets are calculated in business hours. Business hours are 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

Make your description as detailed as possible. Include relevant resource information, along with anything else that might be help us understand your issue. 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 can 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`).

When your case is updated, you receive email with the correspondence and a link to the case in Support Center—you can't respond to case correspondence by email. When you're satisfied with the response or your problem is solved, you can select **Close Case** in Support Center. If you don't respond within six days, the case is closed automatically. You can always reopen a resolved or closed case.

Be sure 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. If you open a case related to old inquiries, include (where possible) 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:

- Use the email address and password associated with your AWS account
- Use AWS Identity and Access Management (Preferred)

Customers with a Business or Enterprise support plan can also access AWS Support and Trusted Advisor operations programmatically by using the AWS Support API (p. 11).

AWS Account

You can use your AWS account information to access Support Center. Sign in at https://console.aws.amazon.com/support/home#/, and then enter your email address and password. However, avoid using this method as much as possible. Instead, use IAM. 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 the cases that are created for the account.

By default, IAM users can't access the Support Center. You can 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 user, group, or role. Support doesn't let you allow or deny access to individual actions. Therefore, the `Action` element of a policy is always set to `support:*`. Similarly, Support doesn't 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 with 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 serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment, and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. All AWS customers have access to five 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. 34).

Customers can access Trusted Advisor in the AWS Management Console. Programmatic access to Trusted Advisor is available with the AWS Support API (p. 11).

Troubleshooting

For answers to common troubleshooting questions, see the AWS Support Knowledge Center.

For Windows, Amazon EC2 offers EC2Rescue, which allows customers to examine their Windows instances to help identify common problems, collect log files, and help AWS Support to troubleshoot your issues. You can also use EC2Rescue 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, arranged by service.

Service	Link
Amazon Web Services	Troubleshooting AWS Signature Version 4 Errors
Amazon AppStream	Troubleshoot Amazon AppStream
Amazon EC2 Auto Scaling	Troubleshooting Auto Scaling
AWS Certificate Manager (ACM)	Troubleshooting
AWS CloudFormation	Troubleshooting AWS CloudFormation
Amazon CloudFront	Troubleshooting | Troubleshooting RTMP Distributions
AWS CloudHSM	Troubleshooting
Amazon CloudSearch	Troubleshooting Amazon CloudSearch
AWS CodeDeploy	Troubleshooting AWS CodeDeploy

Service	Link
AWS Data Pipeline	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 Elastic 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 for Memcached	Troubleshooting Applications
Amazon ElastiCache for Redis	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
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 Route 53	Troubleshooting Amazon Route 53
Amazon Silk	Troubleshooting
Amazon Simple Email Service (Amazon SES)	Troubleshooting Amazon SES

Service	Link
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. 11) 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

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 are resolved.
- Add communications and file attachments to your cases, and add the 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. 31).

For example Java code that demonstrates how to manage the entire life cycle of an AWS Support case, see Programming an AWS Support Case (p. 13).

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. 34).

For an example that uses the Trusted Advisor operations, see Using Trusted Advisor as a Web Service (p. 27).

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. Be sure that you 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 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

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 then 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

To attach files to the case, you must add the attachments to an attachment set before creating the case. You can add up to three attachments to an attachment 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 then 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 that AWS Support provides support in. 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 usually 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 the 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 Service-Linked Roles

AWS Support and AWS Trusted Advisor use AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique IAM role that is linked directly to AWS Support and Trusted Advisor. In each case, the service-linked role is a predefined role. This role includes all the permissions that AWS Support or Trusted Advisor require to call other AWS services on your behalf. The following topics explain what service-linked roles do and how to work with them in AWS Support and Trusted Advisor.

Topics

Using Service-Linked Roles for AWS Support

AWS Support tools gather information about your AWS resources through API calls to provide world-class customer service and technical support. To increase the transparency and auditability of support activities, AWS Support uses an AWS Identity and Access Management (IAM) service-linked role. The `AWSServiceRoleForSupport` service-linked role is a unique IAM role that is linked directly to AWS Support. This service-linked role is predefined, and it includes all the permissions that AWS Support requires to call other AWS services on your behalf. AWS Support uses this service-linked role in various ways:

- **Billing, administrative, support, and other customer services.** As an AWS customer, you automatically have around-the-clock access to AWS customer service. AWS customer service uses the permissions granted by the service-linked role to perform a number of services as part of your support plan. These include investigating and answering account and billing questions, providing administrative support for your account, increasing service limits, and offering additional customer support.
- **Processing of service attributes and usage data for your AWS account.** To provide billing, administrative, and support services, AWS Support might use the permissions granted by the service-linked role to access service attributes and usage data for your AWS account. Service attributes include your account's resource identifiers, metadata tags, roles, and permissions. Usage data includes usage policies, usage statistics, and analytics.
- **Maintaining the operational health of your account and its resources.** AWS Support uses automated tools to perform actions related to operational and technical support.

To provide these services, the role's predefined permissions give AWS Support access to resource metadata, not customer data. Only AWS Support tools can assume this role, which exists within your AWS account.

We redact fields that could contain customer data. For example, the Input and Output fields of the `GetExecutionHistory` for AWS Step Functions API call aren't visible to AWS Support.

For more information about the `AWSServiceRoleForSupport` role or its uses, contact AWS Support.

> **Note**
> AWS Trusted Advisor uses a separate IAM service-linked role for accessing AWS resources for your account to provide best practice recommendations and checks. For more information, see Using Service-Linked Roles for Trusted Advisor (p. 23).

The `AWSServiceRoleForSupport` service-linked role enables all support API calls to be visible to customers through AWS CloudTrail. This helps with monitoring and auditing requirements, because it provides a transparent way to understand the actions that AWS Support performs on your behalf. For information about CloudTrail, see the AWS CloudTrail User Guide.

Service-Linked Role Permissions for AWS Support

The `AWSServiceRoleForSupport` service-linked role trusts the support.amazonaws.com service to assume the role. The permissions policy of the service-linked role contains all the permissions that AWS Support needs to complete actions on your behalf.

For more information about the `AWSServiceRoleForSupport` role or its uses, contact AWS Support.

Creating a Service-Linked Role for AWS Support

You don't need to manually create the `AWSServiceRoleForSupport` role. When you create a new AWS account, this role is automatically created and configured for you.

> **Important**
> If you were using AWS Support before it began supporting service-linked roles, then AWS created the `AWSServiceRoleForSupport` role in your account. To learn more, see A New Role Appeared in My IAM Account.

Editing and Deleting a Service-Linked Role for AWS Support

You can use IAM to edit the description for the `AWSServiceRoleForSupport` service-linked role. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

The `AWSServiceRoleForSupport` role is necessary for AWS Support to provide administrative, operational, and technical support for your account. As a result, this role can't be deleted through the IAM console, API, or CLI. This protects your AWS account, because you can't inadvertently remove necessary permissions for administering support services.

For more information about the `AWSServiceRoleForSupport` role or its uses, contact AWS Support.

Using Service-Linked Roles for Trusted Advisor

AWS Trusted Advisor uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique IAM role that is linked directly to Trusted Advisor. Service-linked roles are predefined by Trusted Advisor, and they include all the permissions that the service requires to call other AWS services on your behalf. Trusted Advisor uses this role to check your usage across AWS and to provide recommendations for improving your AWS environment. For example, Trusted Advisor analyzes your Amazon EC2 instance use to help you reduce costs, increase performance, tolerate failures, and improve security.

This role simplifies getting started with your AWS account, because you don't have to add the necessary permissions for Trusted Advisor. Trusted Advisor defines the permissions of its service-linked role, and only Trusted Advisor can assume this role. The defined permissions include the trust policy and the permissions policy. That permissions policy can't be attached to any other IAM entity.

You can delete the role only after you first disable Trusted Advisor. This prevents you from removing permissions required by Trusted Advisor operations. When you disable Trusted Advisor, you disable all service features, including offline processing and notifications. Also, if you disable Trusted Advisor for a linked account, then the separate payer account is also affected, which negates some cost-saving functionality. You can re-enable Trusted Advisor only after installing the AWSServiceRoleForTrustedAdvisor in the account through IAM.

Note
AWS Support uses a separate IAM service-linked role for accessing your account's resources to provide billing, administrative, and support services. For more information, see Using Service-Linked Roles for AWS Support (p. 22).

For information about other services that support service-linked roles, see AWS Services That Work with IAM. Look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-Linked Role Permissions for Trusted Advisor

Trusted Advisor uses the service-linked role named AWSServiceRoleForTrustedAdvisor—which allows Trusted Advisor to access AWS services on your behalf.

The AWSServiceRoleForTrustedAdvisor service-linked role trusts the following services to assume the role:

- `trustedadvisor.amazonaws.com`

The role permissions policy allows Trusted Advisor to complete the following actions on the specified resources:

- Action: `Read-only access` on `all AWS resources`

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role.

To allow an IAM entity to create the `AWSServiceRoleForTrustedAdvisor` service-linked role

This is necessary only if the Trusted Advisor account is disabled, the service-linked role is deleted, and the user must recreate the role to re-enable Trusted Advisor.

Add the following statement to the permissions policy for the IAM entity that needs to create the service-linked role:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/trustedadvisor.amazonaws.com/
AWSServiceRoleForTrustedAdvisor*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "trustedadvisor.amazonaws.com"}}
}
```

To allow an IAM entity to edit the description of the `AWSServiceRoleForTrustedAdvisor` service-linked role

Due to the nature of this role, only its description can be edited.

Add the following statement to the permissions policy for the IAM entity that needs to edit the description of a service-linked role:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
```

```
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/trustedadvisor.amazonaws.com/
AWSServiceRoleForTrustedAdvisor*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "trustedadvisor.amazonaws.com"}}
}
```

To allow an IAM entity to delete the `AWSServiceRoleForTrustedAdvisor` service-linked role

Add the following statement to the permissions policy for the IAM entity that needs to delete a service-linked role:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/trustedadvisor.amazonaws.com/
AWSServiceRoleForTrustedAdvisor*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "trustedadvisor.amazonaws.com"}}
}
```

You can also use an AWS managed policy, such as AdministratorAccess, to provide full access to Trusted Advisor.

Creating a Service-Linked Role for Trusted Advisor

You don't need to manually create the `AWSServiceRoleForTrustedAdvisor` service-linked role. When you open an AWS account, Trusted Advisor creates the service-linked role for you.

> **Important**
> If you were using the Trusted Advisor service before it began supporting service-linked roles, then Trusted Advisor already created the `AWSServiceRoleForTrustedAdvisor` role in your account. To learn more, see A New Role Appeared in My IAM Account.

If your account has no `AWSServiceRoleForTrustedAdvisor` service-linked role, then Trusted Advisor won't work as expected. This could happen if someone in your account disabled Trusted Advisor and then deleted the service-linked role. In this case, you can use IAM to create the `AWSServiceRoleForTrustedAdvisor` service-linked role, and then enable Trusted Advisor.

To enable Trusted Advisor (console)

1. First, use the IAM console, the IAM CLI, or the IAM API to create a new service-linked role using the Trusted Advisor use case. For more information, see Creating a Service-Linked Role.
2. Sign in to the AWS Management Console, and then open the Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.

   Your Trusted Advisor console experience will be blocked by the **Disabled Trusted Advisor** status banner.
3. Choose **Enable Trusted Advisor Role** from the Disabled status banner. Upon success, the Trusted Advisor console experience is enabled. If the required `AWSServiceRoleForTrustedAdvisor` isn't detected, the Disabled status banner remains.

Editing a Service-Linked Role for Trusted Advisor

Trusted Advisor doesn't allow you to edit the `AWSServiceRoleForTrustedAdvisor` service-linked role if your account has no Trusted Advisor service-linked role. After you create a service-linked role, you can't change the name of the role, because various entities might reference the role. However, you can use the IAM console, the IAM CLI, or the IAM API to edit the description of the role. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a Service-Linked Role for Trusted Advisor

If you don't need to use the features or services of Trusted Advisor, we recommend that you delete the `AWSServiceRoleForTrustedAdvisor` role. Disabling Trusted Advisor and then deleting its service-linked role blocks all Trusted Advisor functionality for the entire account. The Trusted Advisor console will be blocked, and will display the Disabled status. API calls to Trusted Advisor will return an Access Denied error.

Before you can delete the `AWSServiceRoleForTrustedAdvisor` role using IAM, you must first disable Trusted Advisor in the console.

> **Note**
> When you disable Trusted Advisor and delete its service-linked role, some cost-saving functionality within a separate, linked payer account will be negatively affected.)

Cleaning Up a Service-Linked Role

Before you can use IAM to delete a service-linked role, you must first disable Trusted Advisor using the console.

To disable Trusted Advisor (console)

1. Sign in to the AWS Management Console, and then open the Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.
2. In the navigation pane of the Trusted Advisor console, choose **Preferences**.
3. In the **Service Linked Role Permissions** section, choose **Disable Trusted Advisor**.
4. In the confirmation dialog box, confirm that you want to disable Trusted Advisor by choosing **OK**.

When successful, all Trusted Advisor functionality is disabled, and the Trusted Advisor console displays only the Disabled status banner.

You can then use the IAM console, the IAM CLI, or the IAM API to delete the Trusted Advisor service-linked role named `AWSServiceRoleForTrustedAdvisor`. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

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 then get 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

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 AWSSupport createClient()
{
    return AWSSupportClientBuilder.defaultClient();
}

// Get the List of Available Trusted Advisor Checks
public static void getTAChecks() {
    // Possible language parameters: "en" (English), "ja" (Japanese), "fr" (French),
 "zh" (Chinese)
    DescribeTrustedAdvisorChecksRequest request = new
 DescribeTrustedAdvisorChecksRequest().withLanguage("en");

    DescribeTrustedAdvisorChecksResult result =
 createClient().describeTrustedAdvisorChecks(request);

    for (TrustedAdvisorCheckDescription description : result.getChecks()) {
        // Do something with check description.
        System.out.println(description.getId());
        System.out.println(description.getName());
    }
}
```

Refresh 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 refresh Trusted Advisor data.

```
// Refresh a Trusted Advisor Check
// Note: Some checks are refreshed automatically, and they cannot be refreshed by using
 this operation.
// Specifying the check ID of a check that is automatically refreshed causes an
 InvalidParameterValue error.
public static void refreshTACheck(final String checkId) {
    RefreshTrustedAdvisorCheckRequest request = new
 RefreshTrustedAdvisorCheckRequest().withCheckId(checkId);
    RefreshTrustedAdvisorCheckResult result =
 createClient().refreshTrustedAdvisorCheck(request);

    System.out.println("CheckId: " + result.getStatus().getCheckId());
    System.out.println("Milliseconds until refreshable: " +
 result.getStatus().getMillisUntilNextRefreshable());
    System.out.println("Refresh Status: " + result.getStatus().getStatus());
}
```

Poll a Trusted Advisor Check for Status Changes

After you submit the request to run a Trusted Advisor check to generate the latest status data, you use the `DescribeTrustedAdvisorCheckRefreshStatuses` action to request the progress of the check's run, and when new data is ready for the check.

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. Finally, the code returns an instance of a `DescribeTrustedAdvisorCheckResultResult` data type that you can use to traverse the information produced by the check.

Note: Use a single refresh request before polling for the status of the request.

```
// Retrieves TA refresh statuses. Multiple checkId's can be submitted.
public static List<TrustedAdvisorCheckRefreshStatus> getTARefreshStatus(final String...
 checkIds) {
    DescribeTrustedAdvisorCheckRefreshStatusesRequest request =
            new DescribeTrustedAdvisorCheckRefreshStatusesRequest().withCheckIds(checkIds);

    DescribeTrustedAdvisorCheckRefreshStatusesResult result =
            createClient().describeTrustedAdvisorCheckRefreshStatuses(request);

    return result.getStatuses();
}

// Retrieves a TA check status, and checks to see if it has finished processing.
public static boolean isTACheckStatusInTerminalState(final String checkId) {
    // Since we only submitted one checkId to getTARefreshStatus, just retrieve the only
 element in the list.
    TrustedAdvisorCheckRefreshStatus status = getTARefreshStatus(checkId).get(0);
    // Valid statuses are:
    // 1. "none", the check has never been refreshed before.
```

```
    // 2. "enqueued", the check is waiting to be processed.
    // 3. "processing", the check is in the midst of being processed.
    // 4. "success", the check has succeeded and finished processing - refresh data is
 available.
    // 5. "abandoned", the check has failed to process.
    return status.getStatus().equals("abandoned") || status.getStatus().equals("success");
}

// Enqueues a Trusted Advisor check refresh. Periodically polls the check refresh status
 for completion.
public static TrustedAdvisorCheckResult getFreshTACheckResult(final String checkId) throws
 InterruptedException {

    refreshTACheck(checkId);

    while(!isTACheckStatusInTerminalState(checkId)) {
        Thread.sleep(10000);
    }

    return getTACheckResult(checkId);
}

// Retrieves fresh TA check data whenever possible.
// Note: Some checks are refreshed automatically, and they cannot be refreshed by using
 this operation. This method
// is only functional for checks that can be refreshed using the RefreshTrustedAdvisorCheck
 operation.
public static void pollForTACheckResultChanges(final String checkId) throws
 InterruptedException {

    String checkResultStatus = null;
    do {

        TrustedAdvisorCheckResult result = getFreshTACheckResult(checkId);

        if (checkResultStatus != null && !checkResultStatus.equals(result.getStatus())) {
            break;
        }

        checkResultStatus = result.getStatus();

        // The rule refresh has completed, but due to throttling rules the checks may not
 be refreshed again
        // for a short period of time.
        // Since we only submitted one checkId to getTARefreshStatus, just retrieve the
 only element in the list.
        TrustedAdvisorCheckRefreshStatus refreshStatus =
 getTARefreshStatus(checkId).get(0);
        Thread.sleep(refreshStatus.getMillisUntilNextRefreshable());

    } while(true);

    // Signal that a TA check has changed check result status here.
}
```

Request a Trusted Advisor Check Result

After you select the check for the detailed results that you want, 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, After you submit the request to run the snippet 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.

```
// Request a Trusted Advisor Check Result
public static TrustedAdvisorCheckResult getTACheckResult(final String checkId) {
    DescribeTrustedAdvisorCheckResultRequest request = new
 DescribeTrustedAdvisorCheckResultRequest()
            // Possible language parameters: "en" (English), "ja" (Japanese),
 "fr" (French), "zh" (Chinese)
            .withLanguage("en")
            .withCheckId(checkId);

    DescribeTrustedAdvisorCheckResultResult requestResult =
 createClient().describeTrustedAdvisorCheckResult(request);

    return requestResult.getResult();
}
```

Note: Requesting a Trusted Advisor Check Result doesn't generate updated results data.

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 AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Support. CloudTrail captures API calls for AWS Support as events. The calls captured include calls from the AWS Support console and code calls to the AWS Support API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS Support. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**.

Using 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 additional details.

To learn more about CloudTrail, including how to configure and enable it, see the AWS CloudTrail User Guide.

AWS Support Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in AWS Support, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for AWS Support, create a *trail*. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

AWS Support supports logging the following actions as events in CloudTrail log files:

- `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 event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (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. 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 Simple Notification Service 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

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source. It includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates 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 with Amazon CloudWatch Events and Amazon CloudWatch

AWS Trusted Advisor is integrated with the Amazon CloudWatch Events and Amazon CloudWatch services. You can use Amazon CloudWatch Events to detect and react to changes in the status of Trusted Advisor checks. And you can use Amazon CloudWatch to create alarms on Trusted Advisor metrics for check status changes, resource status changes, and service limit utilization. See the following topics for details.

Topics

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 the rules that 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
- Amazon Kinesis streams
- Amazon Simple Queue Service queues
- Built-in targets (CloudWatch alarm actions)
- Amazon Simple Notification Service 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 a 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**. Then, enter 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 prepared to use with this rule. Then, configure any additional options required by that type.
11. Choose **Configure details**.
12. On the **Configure rule details** page, enter a name and description for the rule. To enable the rule as soon as it's created, choose the **State** box.
13. If you're satisfied with the rule, choose **Create rule**.

Creating Trusted Advisor Alarms Using CloudWatch

You can use Amazon CloudWatch to create alarms on Trusted Advisor metrics for check status changes, resource status changes, and service limit utilization. Depending on your requirements, you might create multiple alarms.

Follow the basic procedure described here to create a CloudWatch alarm for Trusted Advisor. Before you create alarms for Trusted Advisor metrics, however, you should do the following:

- Familiarize yourself with metrics and alarms in CloudWatch. For more information, see What Is Amazon CloudWatch?
- Refresh your checks through the Trusted Advisor console or through the AWS Support API.

To create a CloudWatch alarm for Trusted Advisor

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation bar, in the Region selector, choose **US East (N. Virginia) Region**.

3. In the navigation pane, choose **Alarms**.
4. Choose **Create Alarm**.
5. For the **Select Metric**, choose a metric for Trusted Advisor.
6. To select a metric, do one of the following:

    a. In the search box, enter one or more dimension values to filter the metric list.
    b. In the results table, select the check box for the row containing the desired metric.
7. Choose **Next**.
8. Configure the alarm:

    a. Under **Alarm Threshold**, specify a name and description.
    b. For the **ServiceLimitUsage** metric, specify a threshold value between **`0.00`** and **`1.00`**.
    c. For the **RedResources**, **YellowResources**, **GreenChecks**, **RedChecks**, and **YellowChecks** metrics, you can specify a threshold that is any whole number greater than or equal to zero.
    d. Configure your desired behavior for missing data. By default, this is set to **missing**.
    e. Under **Actions**, add a notification list.
9. Choose **Create Alarm**.

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 01, 2018

Change	Description	Date Changed
Using Trusted Advisor as a Web Service	Added updated instructions to refresh Trusted Advisor data after getting list of Trusted Advisor checks.	November 01, 2018
Using Service-Linked Roles	Added new section.	July 11, 2018
Getting Started: Troubleshooting	Added troubleshooting links for Route 53 and AWS Certificate Manager.	September 1, 2017
Case Management Example: Creating a Case	Added a note about the **CC** box for users who have the Basic support plan.	August 1, 2017
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*.