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
  Creating a support case ........................................................................... 2
  Example: Create a case for an Amazon EC2 instance ............................ 3
  Choosing a severity ................................................................................. 5
  Describing your problem ....................................................................... 6
Monitoring and maintaining your case ...................................................... 6
Case history .............................................................................................. 6
Access AWS Support .................................................................................. 6
  AWS account ......................................................................................... 7
  IAM ....................................................................................................... 7
Access to AWS Trusted Advisor ............................................................... 8
Changing your AWS support plan ............................................................. 8
AWS Trusted Advisor ............................................................................... 8
  Organizational view for AWS Trusted Advisor .................................... 9
  Trusted Advisor check removal ......................................................... 25
About the AWS Support API ....................................................................... 27
  Support case management .................................................................... 27
  Trusted Advisor .................................................................................. 27
  Endpoint ............................................................................................... 28
  Support in AWS SDKs .......................................................................... 28
Programming an AWS Support case ......................................................... 29
  Overview ............................................................................................... 29
    Using IAM with the AWS Support API ............................................ 29
    Create an AWS Support client ......................................................... 29
    Discover AWS services and issue severity levels ............................ 30
    Create an attachment set ................................................................. 30
    Create a support case ....................................................................... 31
    Retrieve and update support case communications ....................... 34
    Retrieve all support case information ............................................ 35
    Resolve a support case ..................................................................... 36
Using Trusted Advisor as a web service .................................................. 38
  Get the list of available Trusted Advisor checks ................................. 38
  Refresh the list of available Trusted Advisor checks ...................... 38
  Poll a Trusted Advisor check for status changes .............................. 39
  Request a Trusted Advisor check result .......................................... 40
  Print details of a Trusted Advisor check ........................................... 41
Security .................................................................................................... 42
  Data protection .................................................................................... 42
  Identity and access management ....................................................... 43
    Audience .......................................................................................... 43
    Authenticating with identities ......................................................... 44
    Managing access using policies ...................................................... 45
    How AWS Support works with IAM .............................................. 47
    Identity-based policy examples ...................................................... 48
    Using service-linked roles .............................................................. 50
    Manage access for AWS Trusted Advisor .................................... 54
    Troubleshooting .............................................................................. 58
Incident response .................................................................................... 60
Monitoring AWS Support .......................................................................... 60
  Logging AWS Support API calls with AWS CloudTrail .................... 60
  Monitoring Trusted Advisor checks ................................................. 63
Compliance validation ............................................................................. 65

API Version 2013-04-15
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 24/7 access to customer service, AWS documentation, technical papers, and support forums. For technical support and more resources to plan, deploy, and improve your AWS environment, you can choose a support plan that best aligns with your AWS use case.

To create a case in the AWS Management Console, see Creating a support case (p. 2).

Topics
- Features of AWS Support plans (p. 1)
- Case management (p. 2)
- Monitoring and maintaining your case (p. 6)
- Case history (p. 6)
- Access AWS Support (p. 6)
- Changing your AWS support plan (p. 8)
- AWS Trusted Advisor (p. 8)

Features of AWS Support plans

AWS Support offers four support plans: Basic, Developer, Business, and Enterprise.

The Basic plan is Use with caution of charge and offers support for account and billing questions and service quota increases. The other plans offer an Avoid (rewrite) number of technical support cases with pay-by-the-month pricing and no long-term contracts.

All AWS customers automatically have 24/7 access to these features of the Basic support plan:

- One-on-one responses to account and billing questions
- Support forums
- Service health checks
- Documentation, technical papers, and best practice guides

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

- Best practice guidance
- Building-block architecture support: guidance on how to use AWS products, features, and services together
- AWS Identity and Access Management (p. 7) (IAM) to control user 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) – A feature of AWS Support, which inspects customer environments and identifies opportunities to save money, close security gaps, and improve system reliability and performance.
- The AWS Support API to interact with Support Center and Trusted Advisor. You can use the AWS Support API to automate support case management and Trusted Advisor operations.
• Third-party software support – Help with Amazon Elastic Compute Cloud (Amazon EC2) instance operating systems and configuration. Also, help with the performance of the most popular third-party software components on AWS. Third-party software support isn't available for customers on Basic or Developer support plans.

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

• Application architecture guidance – Contextual guidance on how services fit together to meet your specific use case, workload, or application.
• Infrastructure event management – Short-term engagement with AWS Support to get a deep understanding of your use case. After analysis, provide architectural and scaling guidance for an event.
• Technical account manager – Work with a technical account manager (TAM) for your specific use cases and applications.
• White-glove case routing.
• Management business reviews.

For more information about features and pricing for each support plan, see AWS Support and AWS Support features. Some features, such as 24/7 phone and chat support, aren't available in all languages.

Case management

In the AWS Management Console, you can create three types of customer cases in AWS Support:

• **Account and billing support** cases are available to all AWS customers so you can get help with billing and account-related questions.
• **Service limit increase** requests are also available to all AWS customers. For information about the default service quotas, formerly referred to as 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.

  **Note**
  • If you have the Basic support plan, you can't create a technical support case.
  • To close your account, see Closing an Account in the AWS Billing and Cost Management User Guide.

Creating a support case

**To create a support case**

1. Sign in to the AWS Management Console by using the email address and password linked to your AWS account. To sign in with other credentials, see Access AWS Support (p. 6).
2. In the upper-right corner, choose Support and then choose Support Center.
3. Choose Create case.
4. Choose one of the following options:
   • Account and billing support
   • Service limit increase
   • Technical support
5. Follow the prompts to describe your case.

Example: Create a case for an Amazon EC2 instance

As shown in the following screenshot, this example is a technical support case for an Amazon Elastic Compute Cloud (Amazon EC2) instance.

A. Create case – Choose the type of case to create from the three boxes at the top of the page. In this example, the case type is Technical support.

   Note
   If you have the Basic support plan, you can't create a technical support case.

B. Service – If your question affects multiple services, choose the service that's most applicable. In this example, the service is Elastic Compute Cloud (EC2 - Linux).

C. Category – Choose the category that best fits your use case. In this example, there's trouble connecting to an instance, so Instance Issue is chosen. When you choose a category, links to information that might resolve your problem appear below the Case classification section.

D. 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). 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 more information, see Choosing a severity (p. 5).

   Note
   Based on your category choice, you might be prompted for more information. In this example, you're prompted to enter the Instance ID. As a best practice, enter resource IDs, even when not prompted.
After you specify the case type and classification, you can specify the description and how you want to be contacted.

A. **Subject** – Enter a title that briefly describes your issue. In this example, the subject is *Failed status checks*.

B. **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 information, see Describing your problem (p. 6).

C. **Attachments** – Screenshots and other attachments (less than 5 MB each) can be helpful. In this example, the attached image is a failed status check.

D. **Preferred contact language** – Currently, you can choose English or Japanese.
E. **Contact methods** – Choose a contact method. The options depend 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 choose **Chat** or **Phone**. If you choose **Phone**, you're prompted for a callback number.

F. **Additional contacts** – Enter 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 email address. If you’re signed in with your email address and password, you don’t need to include your email address.

Note
If you have the Basic support plan, the **Additional contacts** box isn’t available. However, the **Operations** contact specified in the **Alternate Contacts** section of the **My Account** page receives copies of the case correspondence, but only for the specific case types of account and billing, and technical.

G. **Submit**. Choose **Submit** when your information is complete and you’re ready to create the case.

## Choosing a severity

You might be inclined to always create a support case at the highest severity that your support plan allows. However, we recommend that you choose the highest severities for cases that can’t be worked around or that directly affect production applications. For information about building your services so that losing single resources doesn’t affect your applications, see the Building Fault-Tolerant Applications on AWS technical paper.

The following table lists the severity levels, response times, and example problems.

Note
You can’t change the severity code for a support case after you create one. If your situation changes, work with the AWS Support associate for your support case.

<table>
<thead>
<tr>
<th>Severity</th>
<th>First-response time</th>
<th>Description and support plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>General guidance</td>
<td>24 hours</td>
<td>You have a general development question, or you want to request a feature. (Developer*, Business, and Enterprise support plans)</td>
</tr>
<tr>
<td>System impaired</td>
<td>12 hours</td>
<td>Non-critical functions of your application are behaving abnormally, or you have a time-sensitive development question. (Developer*, Business, and Enterprise support plans)</td>
</tr>
<tr>
<td>Production system impaired</td>
<td>4 hours</td>
<td>Important functions of your application are impaired or degraded. (Business and Enterprise support plans)</td>
</tr>
<tr>
<td>Production system down</td>
<td>1 hour</td>
<td>Your business is significantly impacted. Important functions of your application aren't available. (Business and Enterprise support plans)</td>
</tr>
<tr>
<td>Business-critical system down</td>
<td>15 minutes</td>
<td>Your business is at risk. Critical functions of your application aren't available. (Enterprise support plan)</td>
</tr>
</tbody>
</table>

* 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, excluding holidays and weekends. This information appears in the **Contact Information** section of the **My Account** page in the AWS Management Console. These
times can vary in countries with multiple time zones. Japanese support is available from 9:00 AM to 6:00 PM.

**Note**

We make every reasonable effort to respond to your initial request within the indicated timeframe. For more information about the scope of support for each AWS Support plan, see [AWS Support features](#).

### Describing your problem

Make your description as detailed as possible. Include relevant resource information, along with anything else that might help us understand your issue. For example, to troubleshoot performance, include timestamps 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. However, you can't respond to case correspondence by email. Use the link in the email to navigate to the support case.

**Note**

You must sign in to the AWS account that submitted the support case. If you sign in as an IAM user, you must have the required permissions to view support cases. For more information, see [Access AWS Support](#).

When you're satisfied with the response or your problem is solved, you can choose **Close Case** in Support Center. If you don't respond within ten 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.

### Access AWS Support

You can access the Support Center by using the following options:

- Use the email address and password associated with your AWS account.
- (Recommended) Use AWS Identity and Access Management (IAM).
If you have a Business or Enterprise support plan, you can also use the AWS Support API (p. 27). The API lets you access AWS Support and Trusted Advisor operations programmatically. For more information, see the AWS Support API Reference.

AWS account

You can sign in to the AWS Management Console and access the Support Center by using your AWS account email address and password. This identity is called the AWS account root user. However, we strongly recommend that you don't use the root user for your everyday tasks, even the administrative ones. Instead, we recommend that you use IAM, which lets you control who can perform certain tasks in your account.

IAM

By default, IAM users can't access the Support Center. You can use IAM to create individual users or groups. Then, you attach IAM policies to these entities, so that they have permission to perform actions and access resources, such as to open Support Center cases and use the AWS Support API.

After you create IAM users, you can give those users individual passwords and an account-specific sign-in page. They can then sign in to your AWS account and work in the Support Center. IAM users who have AWS Support access can see all cases that are created for the account.

For more information, see How IAM users sign in to your AWS account.

The easiest way to grant permissions is to attach the AWS managed policy AWSSupportAccess to the user, group, or role. AWS Support allows action-level permissions to control access to specific AWS Support operations. AWS Support doesn't provide resource-level access, so the Resource element is always set to *.

For example, this policy statement grants an IAM user access to AWS Support. An IAM user with this policy can access all AWS Support operations and resources.

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

This policy statement allows an IAM user to perform all actions in AWS Support except resolve a case.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "support:*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "support:ResolveCase",
      "Resource": "*"
    }
  ]
}
```
If the user or group already has a policy, you can add the AWS Support-specific policy statement to that policy.

**Important**
- If you can't view cases in the Support Center, make sure that you have the required permissions. You might need to contact your IAM administrator. For more information, see Identity and access management for AWS Support (p. 43).

### Access to AWS Trusted Advisor

In the AWS Management Console, a separate trustedadvisor IAM namespace controls access to Trusted Advisor. In the AWS Support API, the support IAM namespace controls access to Trusted Advisor. For more information, see Controlling access to the Trusted Advisor console.

### Changing your AWS support plan

You can change your support plan in the AWS Management Console.

**To change your support plan**

2. On the **Support plans** page, choose **Change plan**.
3. On the **Change support plan** page, choose your **New plan**, review the plan information, and then choose **Change plan**.

For an example video of how to change your support plan, see How do I change my AWS Support plan?

**Note**
- If you have an Enterprise support plan, use the link on the **Change support plan** page to contact AWS Support.

### 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. 63).

You can access Trusted Advisor in the AWS Management Console. For more information about controlling access to the Trusted Advisor console, see Manage access for AWS Trusted Advisor (p. 54).

Programmatic access to Trusted Advisor is available with the AWS Support API (p. 27).

**Topics**
- Organizational view for AWS Trusted Advisor (p. 9)
- Trusted Advisor check removal (p. 25)
Organizational view for AWS Trusted Advisor

Organizational view lets you view Trusted Advisor checks for all accounts in your AWS Organizations. After you enable this feature, you can create reports to aggregate the check results for all linked accounts in your organization. The report includes a summary of check results and information about affected resources for each account. For example, you can use the reports to identify which accounts in your organization are using AWS Identity and Access Management (IAM) with the IAM Use check or whether you have recommended actions for Amazon Simple Storage Service (Amazon S3) buckets with the Amazon S3 Bucket Permissions check.

Topics

- Prerequisites (p. 9)
- Enable organizational view (p. 9)
- Refresh Trusted Advisor checks (p. 10)
- Create organizational view reports (p. 10)
- View the report summary (p. 12)
- Download an organizational view report (p. 13)
- Disable organizational view (p. 17)
- Using IAM policies to allow access to organizational view (p. 18)
- Using other AWS services to view Trusted Advisor reports (p. 20)

Prerequisites

You must meet the following requirements to enable organizational view:

- Your accounts must be members of an AWS Organization.
- Your organization must have all features enabled.
- The master account in your organization must have a Business or Enterprise support plan. See Compare AWS Support plans.
- You must sign in as a user in the master account (or assumed equivalent role). Whether you sign in as an IAM user or an IAM role, you must have a policy with the required permissions. See Using IAM policies to allow access to organizational view (p. 18).

Enable organizational view

After you meet the prerequisites, follow these steps to enable organizational view. After you enable this feature, the following happens:

- Trusted Advisor is enabled as a trusted service in your organization. For more information, see Enabling trusted access with other AWS services in the AWS Organizations User Guide.
- The AWSServiceRoleForTrustedAdvisorReporting service-linked-role is created for you in the master account in your organization. This role includes the permissions that Trusted Advisor needs to call Organizations on your behalf. This service-linked role is locked, and you can't delete it manually. For more information, see Using service-linked roles for Trusted Advisor (p. 51).

You enable organizational view from the Trusted Advisor console.

To enable organizational view

1. Sign in as an administrator in the organization's master account and open the AWS Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.
2. In the navigation pane, choose **Organizational View**.
3. Under **Get Started**, choose **Enable organizational view**.

### Refresh Trusted Advisor checks

Before you create a report for your organization, we recommend that you refresh the statuses of your Trusted Advisor checks. You can download a report without refreshing your Trusted Advisor checks, but your report might not have the latest information.

If you have a Business or Enterprise account, Trusted Advisor automatically refreshes the checks in your account on a weekly basis.

**Note**

If you have accounts in your organization that have a Developer or Basic support plan, a user for those accounts must sign in to the Trusted Advisor console to refresh the checks. You can't refresh checks for all accounts from the organization's master account.

**To refresh Trusted Advisor checks**

2. On the **Dashboard** page, choose the refresh icon. This refreshes all checks in your account.

You can also refresh specific checks in the following ways:

- Use the **RefreshTrustedAdvisorCheck** API operation.
- Choose the refresh icon for individual check categories or checks.

### Create organizational view reports

After you enable organizational view, you can create reports so that you can view Trusted Advisor check results for your organization.

You can create up to 50 reports. If you create reports beyond this quota, Trusted Advisor deletes the earliest report. You can't recover deleted reports.

**To create organizational view reports**

1. Sign in to the organization's master account and open the AWS Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.
2. In the navigation pane, choose **Organizational View**.
3. Choose **Create**.
4. By default, the report includes all AWS Regions, accounts in your organization, check categories, checks, and resource statuses. On the **Create report** page, you can use the filter options to customize your report. For example, you can clear the **All** option for **Account**, and then use the list to specify the individual account IDs to include in the report.
   
   a. Enter a **Name** for the report.
   b. For **Format**, choose **JSON** or **CSV**.
c. For **Region**, specify the AWS Regions or choose **All**.
d. For **Account**, specify the account IDs or choose **All**.
e. For **Check category**, specify the check category or choose **All**.
f. For **Checks**, choose the specific checks for that category or choose **All**.
g. For **Resource Status**, choose the status to filter, such as **warning**, or choose **All**.

5. Choose **Create**.

The amount of time it takes to run the report depends on the number of accounts in the organization and the number of resources in each account. On the **Organizational View** page, you can use the **Runtime** and the timestamp for **Date created** to estimate how long it will take to run future reports.

**Example : Create report filter options**

The following example creates a report for the following:

- Three AWS Regions
- All accounts
- All **Security** checks
- The **Idle Load Balancers** and **High Utilization Amazon EC2 Instances** checks
Notes

- Currently, filtering by organizational units (OUs) is not supported. For more information about OUs, see Managing organizational units in the AWS Organizations User Guide.
- You can't create more than one report at a time unless the current report has been running for more than 6 hours.
- Refresh the page if you don't see the report appear on the page.

View the report summary

After the report is ready, you can view the report summary from the Trusted Advisor console. This lets you quickly view the summary of your check results across your organization.

To view the report summary

1. Sign in to the organization's master account and open the AWS Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.
2. In the left navigation pane, choose Organizational View.
3. Choose the report name.

On the Summary page, view the check statuses for each category. You can choose the Filters Applied list to see the filters specified for your report.
Example: Report summary for an organization

![Summary](image)

Download an organizational view report

After your report is ready, download it from the Trusted Advisor console. The report is a .zip file that contains three files:

- summary.json – Contains a summary of the check results for each check category.
- schema.json – Contains the schema for the specified checks in the report.
- A resources file (.json or .csv) – Contains detailed information about the check statuses for resources in your organization.
To download an organizational view report

1. Sign in to the organization's master account and open the AWS Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.
2. In the navigation pane, choose Organizational View.

The Organizational View page displays the available reports to download.
3. Select the check box for the report.
4. Choose Download and save the file.
5. Unzip the file.
6. Use a text editor to open the .json file or a spreadsheet application to open the .csv file.

Note
You might receive multiple files if your report is 5 MB or larger.

Example: summary.json file

The summary.json file shows the number of accounts in the organization and the statuses of the checks in each category.

Trusted Advisor uses the following color code for check results:
- **Green** – Trusted Advisor doesn't detect an issue for the check.
- **Yellow** – Trusted Advisor detects a possible issue for the check.
- **Red** – Trusted Advisor detects an error and recommends an action for the check.
- **Blue** – Trusted Advisor can't determine the status of the check.

In the following example, two checks are Red, two are Yellow, and two are Green.

```json
{
  "numAccounts": 4,
  "filtersApplied": {
    "accountIds": "All",
    "checkIds": [
      "Qch7DwouX1",
      "zXCkfM1nI3",
      "Yw2K9puPzl",
      "hjLMh88UM8"
    ],
    "categories": [
      "security",
      "fault_tolerance"
    ]
  }
}```
Example: schema.json file

The schema.json file includes the schema for the checks in the report. The following example includes the IDs and properties for the IAM Password Policy (Yw2K9puPzl) and IAM Key Rotation (DqdJqYeRm5) checks.

```json
{
  "Yw2K9puPzl": [
    "Password Policy",
    "Uppercase",
    "Lowercase",
    "Number",
    "Non-alphanumeric",
    "Status",
    "Reason"
  ],
  "DqdJqYeRm5": [
    "Status",
    "IAM User",
    "Access Key",
    "Key Last Rotated",
    "Reason"
  ],
  ...
}
```

Example: resources.csv file

The resources.csv file includes information about resources in the organization. This example shows some of the data columns that appear in the report, such as the following:

- Account ID of the affected account
- The Trusted Advisor check ID
- The resource ID
- Timestamp of the report
- The full name of the Trusted Advisor check
- The Trusted Advisor check category
The resources file only contains entries if a check result exists at the resource level. You might not see checks in the report for the following reasons:

- Some checks, such as **MFA on Root Account**, don’t have resources and won’t appear in the report. Checks without resources appear in the `summary.json` file instead.
- Some checks only show resources if they are **Red** or **Yellow**. If all resources are **Green**, they might not appear in your report.
- If an account isn’t enabled for a service that requires the check, the check might not appear in the report. For example, if you’re not using Amazon Elastic Compute Cloud Reserved Instances in your organization, the Amazon EC2 Reserved Instance Lease Expiration check won’t appear in your report.
- The account hasn’t refreshed check results. This might happen when users with a Basic or Developer support plan sign in to the Trusted Advisor console for the first time. If you have a Business or Enterprise support plan, it can take up to one week from account sign up for users to see check results. For more information, see **Refresh Trusted Advisor checks** (p. 10).
- If only the organization’s master account enabled recommendations for checks, the report won’t include resources for other accounts in the organization.

For the resources file, you can use common software such as Microsoft Excel to open the .csv file format. You can use the .csv file for one-time analysis of all checks across all accounts in your organization. If you want to use your report with an application, you can download the report as a .json file instead.

The .json file format provides more flexibility than the .csv file format for advanced use cases such as aggregation and advanced analytics with multiple datasets. For example, you can use a SQL interface with an AWS service such as Amazon Athena to run queries on your reports. You can also use Amazon QuickSight to create dashboards and visualize your data. For more information, see **Using other AWS services to view Trusted Advisor reports** (p. 20).

**Disable organizational view**

Follow this procedure to disable organizational view. You must sign in to the organization’s master account or assume a role with the required permissions to disable this feature. You can’t disable this feature from another account in the organization.

After you disable this feature, the following happens:

- Trusted Advisor is removed as a trusted service in Organizations.
- The `AWSServiceRoleForTrustedAdvisorReporting` service-linked role is unlocked in the organization’s master account. This means you can delete it manually, if needed.
- You can’t create, view, or download reports for your organization. To access previously created reports, you must reenable organizational view from the Trusted Advisor console. See **Enable organizational view** (p. 9).
To disable organizational view for Trusted Advisor

1. Sign in to the organization's master account and open the AWS Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.
2. In the left navigation, choose Preferences.
3. Under Organizational View, choose Disable organizational view.

After you disable organizational view, Trusted Advisor no longer aggregates checks from other AWS accounts in your organization. However, the AWSServiceRoleForTrustedAdvisorReporting service-linked role remains on the organization's master account until you delete it through the IAM console, IAM API, or AWS Command Line Interface (AWS CLI). For more information, see Deleting a service-linked role in the IAM User Guide.

Using IAM policies to allow access to organizational view

You can use the following AWS Identity and Access Management (IAM) policies to allow users or roles in your account access to organizational view in AWS Trusted Advisor.

Example: Full access to organizational view

The following policy allows full access to the organizational view feature. A user with these permissions can do the following:

- Enable organizational view.
- Create, view, and download reports.

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadStatement",
      "Effect": "Allow",
      "Action": [
        "organizations:ListAccountsForParent",
        "organizations:ListAccounts",
        "organizations:DescribeOrganization",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListAWSServiceAccessForOrganization",
        "trustedadvisor:DescribeAccount",
        "trustedadvisor:DescribeChecks",
        "trustedadvisor:DescribeAccountAccess",
        "trustedadvisor:DescribeOrganization",
        "trustedadvisor:DescribeReports",
        "trustedadvisor:DescribeServiceMetadata",
        "trustedadvisor:DescribeOrganizationAccounts"
      ]
    }
  ]
}
```
Example: Read access to organizational view

The following policy allows read-only access to organizational view for Trusted Advisor. A user with these permissions can only view and download existing reports.

```
{  
  "Version": "2012-10-17",  
  "Statement": [    
    {    
      "Sid": "ReadStatement",   
      "Effect": "Allow",   
      "Action": [       
        "organizations:ListAccountsForParent",       
        "organizations:ListAccounts",       
        "organizations:DescribeOrganization",       
        "organizations:ListOrganizationalUnitsForParent",       
        "organizations:ListAWSServiceAccessForOrganization",       
        "trustedadvisor:DescribeAccount",       
        "trustedadvisor:DescribeAccountAccess",       
        "trustedadvisor:DescribeChecks",       
        "trustedadvisor:DescribeOrganization",       
        "trustedadvisor:DescribeReports"    
      ],       
      "Resource": "*"    
    }    
  ]
}
```

You can also create your own IAM policy. For more information, see Creating IAM Policies in the IAM User Guide.

**Note**

If you enabled AWS CloudTrail in your account, the following roles can appear in your log entries:
• AWSServiceRoleForTrustedAdvisorReporting – The service-linked role (SLR) that Trusted Advisor uses to access accounts in your organization.
• AWSServiceRoleForTrustedAdvisor – The service-linked role that Trusted Advisor uses to access services in your organization.

For more information about SLRs, see Using service-linked roles for Trusted Advisor (p. 51).

**Using other AWS services to view Trusted Advisor reports**

Follow this tutorial to upload and view your data by using other AWS services. In this topic, you create an Amazon Simple Storage Service (Amazon S3) bucket to store your report and an AWS CloudFormation template to create resources in your account. Then, you can use Amazon Athena to analyze or run queries for your report or Amazon QuickSight to visualize that data in a dashboard.

**Prerequisites**

Before you start this tutorial, you must meet the following requirements:

• Sign in as an AWS Identity and Access Management (IAM) user with administrator permissions.
• Use the US East (N. Virginia) AWS Region to quickly set up your AWS services and resources.
• Create an Amazon QuickSight account. For more information, see Getting Started with Data Analysis in Amazon QuickSight in the Amazon QuickSight User Guide.

**Upload the report to Amazon S3**

After you download your `resources.json` report, upload the file to Amazon S3. You must use a bucket in the US East (N. Virginia) Region.

**To upload the report to an Amazon S3 bucket**

2. Use the Region selector and choose the US East (N. Virginia) Region.
3. Open the Amazon S3 console at https://console.aws.amazon.com/s3/.
4. Choose an S3 bucket and copy the name. You use the name in the next procedure.
5. Choose Create folder, enter the name `folder1`, and choose Save.
6. Choose the `folder1`.
7. In `folder1`, choose Upload and choose the `resources.json` file.
8. Choose Next, keep the default options, and then choose Upload.

**Note**

If you upload a new report to this bucket, rename the `.json` files each time you upload them so that you don't override the existing reports. For example, you can add the timestamp to each file, such as `resources-timestamp.json`, `resources-timestamp2.json`, and so on.

**Create your resources using AWS CloudFormation**

After you upload your report to Amazon S3, upload the following YAML template to AWS CloudFormation. This template tells AWS CloudFormation what resources to create for your account so that other services can use the report data in the S3 bucket. The template creates resources for IAM, AWS Lambda, and AWS Glue.
To create your resources with AWS CloudFormation

1. Download the trusted-advisor-reports-template.zip file.
2. Unzip the file.
3. Open the template file in a text editor.
4. For the BucketName and FolderName parameters, replace the values for your-bucket-name-here and folder1 with the bucket name and folder name in your account.
5. Save the file.
7. If you haven't already, in the Region selector, choose the US East (N. Virginia) Region.
8. In the navigation pane, choose Stacks.
9. Choose Create stack and choose With new resources (standard).
10. On the Create stack page, under Specify template, choose Upload a template file, and then choose Choose file.
11. Choose the YAML file and choose Next.
12. On the Specify stack details page, enter a stack name such as Organizational-view-Trusted-Advisor-reports, and choose Next.
13. On the Configure stack options page, keep the default options, and then choose Next.
14. On the Review Organizational-view-Trusted-Advisor-reports page, review your options. At the bottom of the page, select the check box for I acknowledge that AWS CloudFormation might create IAM resources.
15. Choose Create stack.
   The stack takes about 5 minutes to create.
16. After the stack creates successfully, the Resources tab appears like the following example.

Query the data in Amazon Athena

After you have your resources, you can view the data in Athena. Use Athena to create queries and analyze the results of the report, such as looking up specific check results for accounts in the organization.

To query the data in Athena

2. If you haven't already, in the Region selector, choose the US East (N. Virginia) Region.

3. Choose Saved Queries and in search field, enter Show sample.

4. Choose the query that appears, such as Show sample entries of TA report.

The query should look like the following.

```
SELECT * FROM "athenatacfn"."folder1" limit 10
```


Example : Athena query

The following example shows 10 sample entries from the report.
Create a dashboard in Amazon QuickSight

You can also set up Amazon QuickSight so that you can view your data in a dashboard and visualize your report information.

**Note**
You must use the US East (N. Virginia) Region.

**To create a dashboard in Amazon QuickSight**

1. Navigate to the Amazon QuickSight console and sign in to your account at https://us-east-1.quicksight.aws.amazon.com.
2. Choose New analysis, New dataset, and then choose Athena.
3. In the New Athena data source dialog box, enter a data source name such as AthenaTA, and then choose Create data source.
4. In the **Choose your table** dialog box, choose the **athenatacfn** table, choose **folder1**, and then choose **Select**.

5. In the **Finish data set creation** dialog box, choose **Directly query your data**, and then choose **Visualize**.

You can now create a dashboard in Amazon QuickSight. For more information, see [Working with Dashboards](#) in the *Amazon QuickSight User Guide*.

**Example : Amazon QuickSight dashboard**

The following example dashboard shows information about the Trusted Advisor checks, such as the following:

- Affected account IDs
- Summary by AWS Regions
- Check categories
• Check statuses
• Number of entries in the report for each account

Note
If you have permission errors while creating your dashboard, make sure that Amazon QuickSight can use Athena. For more information, see I Can't Connect to Amazon Athena in the Amazon QuickSight User Guide.

Trusted Advisor check removal

Important
Trusted Advisor will remove the following checks on November 18, 2020.

<table>
<thead>
<tr>
<th>Checks to be removed on November 18, 2020</th>
<th>Check category</th>
<th>Check ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC2Config Service for EC2 Windows Instances</td>
<td>Fault Tolerance</td>
<td>V77iOLLBqz</td>
</tr>
<tr>
<td>ENA Driver Version for EC2 Windows Instances</td>
<td>Fault Tolerance</td>
<td>TyfdMXG69d</td>
</tr>
<tr>
<td>NVMe Driver Version for EC2 Windows Instances</td>
<td>Fault Tolerance</td>
<td>yHAGQJV9K5</td>
</tr>
<tr>
<td>PV Driver Version for EC2 Windows Instances</td>
<td>Fault Tolerance</td>
<td>Wnwm9II5bG</td>
</tr>
<tr>
<td>EBS Active Volumes</td>
<td>Service Limits</td>
<td>fH7LL0I7J9</td>
</tr>
</tbody>
</table>

If you use results from all checks in the Trusted Advisor console or the AWS Support API, these checks won't appear in check results. If you use any of these five checks such as specifying the check ID in an AWS Support API operation or your code, you will need to remove these checks to avoid API call errors.
Amazon Elastic Block Store no longer has a limit on the number of volumes that you can provision.

You can monitor your Amazon EC2 instances and ensure they are up-to-date by using AWS Systems Manager Distributor, other third-party tools, or write your own scripts to return driver information for Windows Management Instrumentation (WMI).

For more information about the available checks, see the AWS Trusted Advisor best practice checklist.
The AWS Support API provides access to some of the features of the AWS Service Catalog. 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. 27) operations to manage the entire life cycle of your AWS support cases, from creating a case to resolving it.
- Trusted Advisor (p. 27) 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. 27)
- Trusted Advisor (p. 27)
- Endpoint (p. 28)
- Support in AWS SDKs (p. 28)

**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. 60).

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. 29).

**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. 63).

For an example that uses the Trusted Advisor operations, see Using Trusted Advisor as a web service (p. 38).

**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 lets you create cases and add correspondence to them throughout investigations of your issues and interactions with AWS Support staff. This topic demonstrates the use of actions in the AWS Support service, which models much of the behavior of the AWS Support center.

For a list of actions and parameters that you can use for AWS Support, see the AWS Support API Reference.

Topics

- Overview (p. 29)
- Create an AWS Support client (p. 29)
- Discover AWS services and issue severity levels (p. 30)
- Create an attachment set (p. 30)
- Create a support case (p. 31)
- Retrieve and update support case communications (p. 34)
- Retrieve all support case information (p. 35)
- Resolve a support case (p. 36)

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 Access 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.

```java
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:

```java
// 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 the CreateCase action. 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.
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 that you obtained by calling the `DescribeServices` action, 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 action 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 following 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.

```java
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);
    }

    String subject = null;
    String communicationBody = null;
    String severityCode = null;
    String attachmentSetId = null;

    CreateCaseRequest request =
        new CreateCaseRequest(servicecode, categorycode, language, subject, communicationBody, severityCode, attachmentSetId);

    try {
        CaseId caseId = client.createCase(request);
    } catch (AWSServiceException e) {
        e.printStackTrace();
        System.exit(1);
    }
}
```
Create a support case

```java
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)
```

API Version 2013-04-15

33
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 actions to retrieve this correspondence, and the AddAttachmentsToSet and AddCommunicationToCase actions to update the case. These actions 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.

```java
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, which lets you 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.

```java
public static void getCases(AWSSupportClient client)
```
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.");
```

**Note**
The DescribeCases action takes parameters that let you control the number of cases, types of cases, and amount of detail to retrieve. For more information, see the DescribeCases action.
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 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 (p. 38)
- Refresh the list of available Trusted Advisor checks (p. 38)
- Poll a Trusted Advisor check for status changes (p. 39)
- Request a Trusted Advisor check result (p. 40)
- Print details of a Trusted Advisor check (p. 41)

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 API operations. Next, the code gets the list of Trusted Advisor checks and their corresponding CheckId values by calling the DescribeTrustedAdvisorChecks API operation. You can use this information to build user interfaces that enable users to select the check they want to run or refresh.

```java
private static AWSSupport createClient() {
    return AWSSupportClientBuilder.defaultClient();
}

// Get the List of Available Trusted Advisor Checks
public static void getTAChecks() {
    DescribeTrustedAdvisorChecksRequest request = new
    DescribeTrustedAdvisorChecksRequest().withLanguage("en");
    DescribeTrustedAdvisorChecksResult result =
    createClient().describeTrustedAdvisorChecks(request);
    for (TrustedAdvisorCheckDescription description : result.getChecks()) {
        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.

```java
// Refresh a Trusted Advisor Check
```
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 API operation 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.
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 API operation.

Tip
The names and descriptions for Trusted Advisor checks are subject to change. We recommend that you specify the check ID in your code to uniquely identify a check. You can use the DescribeTrustedAdvisorChecks API operation to get the check ID.

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.

```java
// Request a Trusted Advisor Check Result
public static TrustedAdvisorCheckResult getTACheckResult(final String checkId) {
    DescribeTrustedAdvisorCheckResultRequest request = new DescribeTrustedAdvisorCheckResultRequest()
    DescribeTrustedAdvisorCheckResultResult checkResult =
```
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.

```java
// Print ResourceIds for flagged resources.
for (TrustedAdvisorResourceDetail flaggedResource : result1.getResult().getFlaggedResources())
{
    System.out.println(
        "The resource for this ResourceID has been flagged: " +
        flaggedResource.getResourceId());
}
```

**Note:** Requesting a Trusted Advisor Check Result doesn't generate updated results data.
Security in AWS Support

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS compliance programs. To learn about the compliance programs that apply to AWS Support, see AWS services in scope by compliance program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company’s requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using AWS Support. The following topics show you how to configure AWS Support to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS Support resources.

**Topics**
- Data protection in AWS Support (p. 42)
- Identity and access management for AWS Support (p. 43)
- Incident response (p. 60)
- Logging and Monitoring in AWS Support (p. 60)
- Compliance validation for AWS Support (p. 65)
- Resilience in AWS Support (p. 66)
- Infrastructure security in AWS Support (p. 66)
- Configuration and vulnerability analysis in AWS Support (p. 66)

Data protection in AWS Support

AWS Support conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all the AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
• Use SSL/TLS to communicate with AWS resources.

• Set up API and user activity logging with AWS CloudTrail.
• Use AWS encryption solutions, along with all default security controls within AWS services.
• Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you work with AWS Support or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into AWS Support or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

For more information about data protection, see the AWS shared responsibility model and GDPR blog post on the **AWS Security Blog**.

AWS Support gathers and accesses data exclusively over a TLS-protected channel.

## Identity and access management for AWS Support

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be **authenticated** (signed in) and **authorized** (have permissions) to use AWS Support resources. IAM is an AWS service that you can use with no additional charge.

### Topics

- Audience (p. 43)
- Authenticating with identities (p. 44)
- Managing access using policies (p. 45)
- How AWS Support works with IAM (p. 47)
- AWS Support identity-based policy examples (p. 48)
- Using service-linked roles (p. 50)
- Manage access for AWS Trusted Advisor (p. 54)
- Troubleshooting AWS Support identity and access (p. 58)

### Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work you do in AWS Support.

**Service user** – If you use the AWS Support service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more AWS Support features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in AWS Support, see [Troubleshooting AWS Support identity and access](#) (p. 58).

**Service administrator** – If you're in charge of AWS Support resources at your company, you probably have full access to AWS Support. It's your job to determine which AWS Support features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts.
of IAM. To learn more about how your company can use IAM with AWS Support, see How AWS Support works with IAM (p. 47).

IAM administrator – If you’re an IAM administrator, you might want to learn details about how you can write policies to manage access to AWS Support. To view example AWS Support identity-based policies that you can use in IAM, see AWS Support identity-based policy examples (p. 48).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see The IAM Console and Sign-in Page in the IAM User Guide.

You must be authenticated (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication, or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email or your IAM user name. You can access AWS programmatically using your root user or IAM user access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don’t use AWS tools, you must sign the request yourself. Do this using Signature Version 4, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 Signing Process in the AWS General Reference.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Using Multi-Factor Authentication (MFA) in AWS in the IAM User Guide.

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An IAM user is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see Managing Access Keys for IAM Users in the IAM User Guide. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An IAM group is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named IAMAdmins and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to Create an IAM User (Instead of a Role) in the IAM User Guide.
I AM roles

An **I AM role** is an identity within your AWS account that has specific permissions. It is similar to an I AM user, but is not associated with a specific person. You can temporarily assume an I AM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using I AM Roles in the I AM User Guide.

I AM roles with temporary credentials are useful in the following situations:

- **Temporary I AM user permissions** – An I AM user can assume an I AM role to temporarily take on different permissions for a specific task.

- **Federated user access** – Instead of creating an I AM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated Users and Roles in the I AM User Guide.

- **Cross-account access** – You can use an I AM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see How I AM Roles Differ from Resource-based Policies in the I AM User Guide.

- **AWS service access** – A service role is an I AM role that a service assumes to perform actions in your account on your behalf. When you set up some AWS service environments, you must define a role for the service to assume. This service role must include all the permissions that are required for the service to access the AWS resources that it needs. Service roles vary from service to service, but many allow you to choose your permissions as long as you meet the documented requirements for that service. Service roles provide access only within your account and cannot be used to grant access to services in other accounts. You can create, modify, and delete a service role from within I AM. For example, you can create a role that allows Amazon Redshift to access an Amazon S3 bucket on your behalf and then load data from that bucket into an Amazon Redshift cluster. For more information, see Creating a Role to Delegate Permissions to an AWS Service in the I AM User Guide.

- **Applications running on Amazon EC2** – You can use an I AM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an I AM Role to Grant Permissions to Applications Running on Amazon EC2 Instances in the I AM User Guide.

To learn whether to use I AM roles, see When to Create an I AM Role (Instead of a User) in the I AM User Guide.

Managing access using policies

You control access in AWS by creating policies and attaching them to I AM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an entity (root user, I AM user, or I AM role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON Policies in the I AM User Guide.

An I AM administrator can use policies to specify who has access to AWS resources, and what actions they can perform on those resources. Every I AM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission
to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

### Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, role, or group. These policies control what actions that identity can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM Policies in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest_USER_GUIDE/).

Identity-based policies can be further categorized as inline policies or managed policies. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing Between Managed Policies and Inline Policies in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest_USER_GUIDE/).

### Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions Boundaries for IAM Entities in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest_USER_GUIDE/).

- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs Work in the [AWS Organizations User Guide](https://docs.aws.amazon.com/organizations/latest/userguide/).

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session Policies in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest_USER_GUIDE/).

### Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy Evaluation Logic in the [IAM User Guide](https://docs.aws.amazon.com/IAM/latest_USER_GUIDE/).
How AWS Support works with IAM

Before you use IAM to manage access to AWS Support, you should understand what IAM features are available to use with AWS Support. To get a high-level view of how AWS Support and other AWS services work with IAM, see AWS services that work with IAM in the IAM User Guide.

Topics
- AWS Support identity-based policies (p. 47)
- AWS Support IAM roles (p. 47)

AWS Support identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. AWS Support supports specific actions. To learn about the elements that you use in a JSON policy, see IAM JSON policy elements reference in the IAM User Guide.

Actions

The Action element of an IAM identity-based policy describes the specific action or actions that will be allowed or denied by the policy. Policy actions usually have the same name as the associated AWS API operation. The action is used in a policy to grant permissions to perform the associated operation.

Policy actions in AWS Support use the following prefix before the action: support:. For example, to grant someone permission to run an Amazon EC2 instance with the Amazon EC2 RunInstances API operation, you include the ec2:RunInstances action in their policy. Policy statements must include either an Action or NotAction element. AWS Support defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [ 
  "ec2:action1",
  "ec2:action2"
]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word Describe, include the following action:

```
"Action": "ec2:Describe***"
```

To see a list of AWS Support actions, see Actions Defined by AWS Support in the IAM User Guide.

Examples

To view examples of AWS Support identity-based policies, see AWS Support identity-based policy examples (p. 48).

AWS Support IAM roles

An IAM role is an entity within your AWS account that has specific permissions.

Using temporary credentials with AWS Support

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as AssumeRole or GetFederationToken.
AWS Support supports using temporary credentials.

Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

AWS Support supports service-linked roles. For details about creating or managing AWS Support service-linked roles, see Using service-linked roles for AWS Support (p. 50).

Service roles

This feature allows a service to assume a service role on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

AWS Support supports service roles.

AWS Support identity-based policy examples

By default, IAM users and roles don't have permission to create or modify AWS Support resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating policies on the JSON tab in the IAM User Guide.

Topics

- Policy best practices (p. 48)
- Using the AWS Support console (p. 49)
- Allow users to view their own permissions (p. 49)

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete AWS Support resources in your account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- Get Started Using AWS Managed Policies – To start using AWS Support quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the IAM User Guide.

- Grant Least Privilege – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the IAM User Guide.

- Enable MFA for Sensitive Operations – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the IAM User Guide.
• **Use Policy Conditions for Extra Security** – To the extent that it's practical, define the conditions under
which your identity-based policies allow access to a resource. For example, you can write conditions to
specify a range of allowable IP addresses that a request must come from. You can also write conditions
to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For
more information, see [IAM JSON policy elements: Condition](https://docs.aws.amazon.com/iam/latest/userguide/)
in the *IAM User Guide*.

### Using the AWS Support console

To access the AWS Support console, you must have a minimum set of permissions. These permissions
must allow you to list and view details about the AWS Support resources in your AWS account. If you
create an identity-based policy that is more restrictive than the minimum required permissions, the
console won't function as intended for entities (IAM users or roles) with that policy.

To be sure that those entities can still use the AWS Support console, also attach the following AWS
managed policy to the entities. For more information, see [Adding permissions to a user](https://docs.aws.amazon.com/iam/latest/userguide/)
in the *IAM User Guide*:

You don't need to allow minimum console permissions for users that are making calls only to the AWS
CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're
trying to perform.

### Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed
policies that are attached to their user identity. This policy includes permissions to complete this action
on the console or programmatically using the AWS CLI or AWS API.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"],
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*
        }
    ]
}
```
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 (p. 50)
- Using service-linked roles for Trusted Advisor (p. 51)

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. 51).

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 the AWS Identity and Access Management (IAM) service-linked role. A service-linked role is a unique IAM role that is linked directly to AWS 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 to improve your AWS environment. For example, Trusted Advisor analyzes your Amazon Elastic Compute Cloud (Amazon EC2) instance use to help you reduce costs, increase performance, tolerate failures, and improve security.

**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. 50).

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.

**Topics**

- Service-linked role permissions for Trusted Advisor (p. 52)
- Manage permissions for service-linked roles (p. 52)
- Creating a service-linked role for Trusted Advisor (p. 53)
- Editing a service-linked role for Trusted Advisor (p. 53)
- Deleting a service-linked role for Trusted Advisor (p. 53)
Service-linked role permissions for Trusted Advisor

Trusted Advisor uses two service-linked roles:

- **AWSServiceRoleForTrustedAdvisor** – This role trusts the Trusted Advisor service to assume the role to access AWS services on your behalf. The role permissions policy allows Trusted Advisor read-only access for all AWS resources. This role simplifies getting started with your AWS account, because you don't have to add the necessary permissions for Trusted Advisor. When you open an AWS account, Trusted Advisor creates this role for you. The defined permissions include the trust policy and the permissions policy. You can't attach the permissions policy to any other IAM entity.

- **AWSServiceRoleForTrustedAdvisorReporting** – This role trusts the Trusted Advisor service to assume the role for the organizational view feature. This role enables Trusted Advisor as a trusted service in your AWS Organizations organization. Trusted Advisor creates this role for you when you enable organizational view. Use this feature to create reports for Trusted Advisor check results for all accounts in your organization. For more information, see Organizational view for AWS Trusted Advisor (p. 9).

Manage permissions for service-linked roles

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. The following examples use the **AWSServiceRoleForTrustedAdvisor** service-linked role.

**Example: Allow an IAM entity to create the AWSServiceRoleForTrustedAdvisor service-linked role**

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

You can add the following statement to the permissions policy for the IAM entity to create the service-linked role.

```json
{
  "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"}}
}
```

**Example: Allow an IAM entity to edit the description of the AWSServiceRoleForTrustedAdvisor service-linked role**

You can only edit the description for the **AWSServiceRoleForTrustedAdvisor** role. You can add the following statement to the permissions policy for the IAM entity to edit the description of a service-linked role.

```json
{
  "Effect": "Allow",
  "Action": [
    "iam:UpdateRoleDescription"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/trustedadvisor.amazonaws.com/AWSServiceRoleForTrustedAdvisor*",
  "Condition": {"StringLike": {"iam:AWSServiceName": "trustedadvisor.amazonaws.com"}}
}
```
Example: Allow an IAM entity to delete the AWSServiceRoleForTrustedAdvisor service-linked role

You can add the following statement to the permissions policy for the IAM entity 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 in the IAM User Guide.

If your account doesn't have the AWSServiceRoleForTrustedAdvisor service-linked role, then Trusted Advisor won't work as expected. This can 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 reenable Trusted Advisor.

To enable Trusted Advisor (console)

1. Use the IAM console, AWS CLI, or the IAM API to create a service-linked role for Trusted Advisor. For more information, see Creating a service-linked role.
2. Sign in to the AWS Management Console, and then navigate to the Trusted Advisor console at https://console.aws.amazon.com/trustedadvisor.
   
   The Disabled Trusted Advisor status banner appears in the console.
3. Choose Enable Trusted Advisor Role from the status banner. If the required AWSServiceRoleForTrustedAdvisor isn't detected, the disabled status banner remains.

Editing a service-linked role for Trusted Advisor

You can't change the name of a service-linked role because various entities might reference the role. However, you can use the IAM console, AWS 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, you can delete the AWSServiceRoleForTrustedAdvisor role. You must disable Trusted Advisor before you can delete this service-linked role. 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 means you won't receive Trusted Advisor checks that identify ways to save costs. You can't access the Trusted Advisor console. API calls to Trusted Advisor return an access denied error.

You must recreate the AWSServiceRoleForTrustedAdvisor service-linked role in the account before you can reenable Trusted Advisor.

You must first disable Trusted Advisor in the console before you can delete the AWSServiceRoleForTrustedAdvisor service-linked role.

**To disable Trusted Advisor**

2. In the navigation pane, choose Preferences.
3. In the Service Linked Role Permissions section, choose Disable Trusted Advisor.
4. In the confirmation dialog box, choose OK to confirm that you want to disable Trusted Advisor.

After you disable Trusted Advisor, 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 AWS 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.

**Manage access for AWS Trusted Advisor**

You can access AWS Trusted Advisor from the AWS Management Console. All AWS accounts have access to a select core Trusted Advisor checks. If you have a Business or Enterprise support plan, you can access all Trusted Advisor checks.

You can use AWS Identity and Access Management (IAM) to control access to Trusted Advisor.

**Topics**

- Permissions for the Trusted Advisor console (p. 54)
- Trusted Advisor actions (p. 55)
- IAM policy examples (p. 56)
- See also (p. 58)

**Permissions for the Trusted Advisor console**

You must have a minimum set of permissions to access the Trusted Advisor console. These permissions must allow you to list and view details about the Trusted Advisor resources in your AWS account.

You can use the following options to control access to Trusted Advisor:

- Use the tag filter feature of the Trusted Advisor console. The user or role must have permissions associated with the tags.

  You can use AWS managed policies or custom policies to assign permissions by tags. For more information, see Obtaining Permissions for Tagging.

- Create an IAM policy with the trustedadvisor namespace, so that you can specify permissions to actions and resources.
When you create a policy, you can specify the namespace of the service to allow or deny an action. The namespace for Trusted Advisor is `trustedadvisor`.

**Important**
You can't use the `trustedadvisor` namespace to allow or deny Trusted Advisor API operations in the AWS Support API. The namespace for AWS Support is `support`.

## Trusted Advisor actions

You can perform the following Trusted Advisor actions in the console. You can also specify these Trusted Advisor actions in an IAM policy to allow or deny specific actions.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DescribeAccount</td>
<td>Grants permission to view the AWS Support plan and various Trusted Advisor preferences.</td>
</tr>
<tr>
<td>DescribeAccountAccess</td>
<td>Grants permission to view if the AWS account has enabled or disabled Trusted Advisor.</td>
</tr>
<tr>
<td>DescribeCheckItems</td>
<td>Grants permission to view details for the check items.</td>
</tr>
<tr>
<td>DescribeCheckRefreshStatuses</td>
<td>Grants permission to view the refresh statuses for Trusted Advisor checks.</td>
</tr>
<tr>
<td>DescribeCheckSummaries</td>
<td>Grants permission to view Trusted Advisor check summaries.</td>
</tr>
<tr>
<td>DescribeChecks</td>
<td>Grants permission to view details for Trusted Advisor checks.</td>
</tr>
<tr>
<td>DescribeNotificationPreferences</td>
<td>Grants permission to view the notification preferences for the AWS account.</td>
</tr>
<tr>
<td>ExcludeCheckItems</td>
<td>Grants permission to exclude recommendations for Trusted Advisor checks.</td>
</tr>
<tr>
<td>IncludeCheckItems</td>
<td>Grants permission to include recommendations for Trusted Advisor checks.</td>
</tr>
<tr>
<td>RefreshCheck</td>
<td>Grants permission to refresh a Trusted Advisor check.</td>
</tr>
<tr>
<td>SetAccountAccess</td>
<td>Grants permission to enable or disable Trusted Advisor for the account.</td>
</tr>
<tr>
<td>UpdateNotificationPreferences</td>
<td>Grants permission to update notification preferences for Trusted Advisor.</td>
</tr>
</tbody>
</table>

The following Trusted Advisor actions are for the organizational view feature. For more information, see [Organizational view for AWS Trusted Advisor](#).

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DescribeOrganization</td>
<td>Grants permission to view if the AWS account meets the requirements to enable the organizational view feature.</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DescribeOrganizationAccounts</td>
<td>Grants permission to view the linked AWS accounts that are in the organization.</td>
</tr>
<tr>
<td>DescribeReports</td>
<td>Grants permission to view details for organizational view reports, such as the report name, runtime, date created, status, and format.</td>
</tr>
<tr>
<td>DescribeServiceMetadata</td>
<td>Grants permission to view information about organizational view reports, such as the AWS Regions, check categories, check names, and resource statuses.</td>
</tr>
<tr>
<td>GenerateReport</td>
<td>Grants permission to create a report for Trusted Advisor checks in your organization.</td>
</tr>
<tr>
<td>SetOrganizationAccess</td>
<td>Grants permission to enable the organizational view feature for Trusted Advisor.</td>
</tr>
</tbody>
</table>

IAM policy examples

The following policies show you how to allow and deny access to Trusted Advisor.

Examples

- Full access to Trusted Advisor (p. 56)
- Read-only access to Trusted Advisor (p. 56)
- Deny access to Trusted Advisor (p. 57)
- Allow and deny specific actions (p. 57)
- Control access to the AWS Support API operations for Trusted Advisor (p. 57)

Full access to Trusted Advisor

The following policy allows users to view and take all actions on all Trusted Advisor checks.

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

Read-only access to Trusted Advisor

The following policy allows users read-only access to Trusted Advisor. Users can't make changes, such as refresh checks or change notification preferences.

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

Deny access to Trusted Advisor

The following policy doesn't allow users to view or take actions for Trusted Advisor checks.

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

Allow and deny specific actions

The following policy allows users to view all Trusted Advisor checks, but doesn't allow them to refresh any checks.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "trustedadvisor:*",
         "Resource": "*"
      },
      {
         "Effect": "Deny",
         "Action": "trustedadvisor:RefreshCheck",
         "Resource": "*"
      }
   ]
}
```

Control access to the AWS Support API operations for Trusted Advisor

In the AWS Management Console, a separate trustedadvisor IAM namespace controls access to Trusted Advisor. You can't use the trustedadvisor namespace to allow or deny Trusted Advisor API operations in the AWS Support API. Instead, you use the support IAM namespace. You must have permissions to the AWS Support API to call Trusted Advisor programmatically.

For example, if you want to call the RefreshTrustedAdvisorCheck operation, you must have permissions to this action in the policy.

The following policy allows users only access to the AWS Support API operations for Trusted Advisor, but not the rest of the AWS Support API operations.

```json
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "support:*",
         "Resource": "*"
      },
      {
         "Effect": "Deny",
         "Action": "support:*",
         "Resource": "*"
      }
   ]
}
```
For more information about how IAM works with AWS Support and Trusted Advisor, see Actions (p. 47).

See also

For more information about Trusted Advisor permissions, see the following resources:

- Actions defined by AWS Trusted Advisor in the IAM User Guide.
- Controlling Access to the Trusted Advisor Console

Troubleshooting AWS Support identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with AWS Support and IAM.

Topics

- I'm not authorized to perform iam:PassRole (p. 58)
- I want to view my access keys (p. 59)
- I'm an administrator and want to allow others to access AWS Support (p. 59)
- I want to allow people outside of my AWS account to access my AWS Support resources (p. 59)

I'm not authorized to perform iam:PassRole

If you receive an error that you’re not authorized to perform the iam:PassRole action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to AWS Support.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.
The following example error occurs when an IAM user named marymajor tries to use the console to perform an action in AWS Support. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

**I want to view my access keys**

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, AKIAIOSFODNN7EXAMPLE) and a secret access key (for example, wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

**Important**

Do not provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see Managing Access Keys in the *IAM User Guide*.

**I'm an administrator and want to allow others to access AWS Support**

To allow others to access AWS Support, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in AWS Support.

To get started right away, see Creating Your First IAM Delegated User and Group in the *IAM User Guide*.

**I want to allow people outside of my AWS account to access my AWS Support resources**

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether AWS Support supports these features, see How AWS Support works with IAM (p. 47).
- To learn how to provide access to your resources across AWS accounts that you own, see Providing Access to an IAM User in Another AWS Account That You Own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing Access to AWS Accounts Owned by Third Parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing Access to Externally Authenticated Users (Identity Federation) in the *IAM User Guide*.
Incident response

Incident response for AWS Support is an AWS responsibility. AWS has a formal, documented policy and program that governs incident response. For more information, see the Introducing the AWS Security Incident Response Whitepaper.

Use the following options to inform yourself about operational issues:

- View AWS operational issues with broad impact on the AWS Service Health Dashboard. For example, events that affect a service or Region that isn't specific to your account.
- View operational issues for individual accounts in the Personal Health Dashboard. For example, events that affect services or resources in your account. For more information, see Getting started with the AWS Personal Health Dashboard in the AWS Health User Guide.

Logging and Monitoring in AWS Support

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS Support and your other AWS solutions. AWS provides the following monitoring tools to watch AWS Support, report when something is wrong, and take automatic actions when appropriate:

- Amazon CloudWatch monitors your AWS resources and the applications you run on AWS in real time. You can collect and track metrics, create customized dashboards, and set alarms that notify you or take actions when a specified metric reaches a threshold that you specify. For example, you can have CloudWatch track CPU usage or other metrics of your Amazon EC2 instances and automatically launch new instances when needed. For more information, see the Amazon CloudWatch User Guide.
- Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources. CloudWatch Events enables automated event-driven computing, as you can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see the Amazon CloudWatch Events User Guide.
- AWS CloudTrail captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Topics

- Logging AWS Support API calls with AWS CloudTrail (p. 60)
- Monitoring Trusted Advisor checks (p. 63)

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 information in CloudTrail logging**

When CloudTrail logging is enabled in your AWS account, API calls made to specific AWS Support operations are tracked in CloudTrail log files. AWS Support operations 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 operations are supported:

- AddAttachmentsToSet
- AddCommunicationToCase
- CreateCase
- DescribeAttachment
- DescribeCases
- DescribeCommunications
- DescribeServices
- DescribeSeverityLevels
- ResolveCase

CloudTrail doesn't support logging for the AWS Support API operations for AWS Trusted Advisor, such as DescribeTrustedAdvisorChecks. For more information about the AWS Support API operations, see the AWS Support API Reference.

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.

**Trusted Advisor information in CloudTrail logging**

AWS Trusted Advisor is a feature in AWS Support that lets you check your AWS account for ways to save costs, improve security, and optimize your AWS account.

CloudTrail doesn't log Trusted Advisor operations.

For a list of supported Trusted Advisor operations, see Trusted Advisor actions (p. 55).

**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 operation, the date and time of the operation, 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 operation.

```json
{
    "Records": [
        {
            "eventVersion": "1.04",
            "userIdentity": {
                "type": "IAMUser",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/janedoe",
                "accountId": "111122223333",
                "accessKeyId": "AKIAIOSFODDN7EXAMPLE",
                "userName": "janedoe",
                "sessionContext": {
                    "attributes": {
                        "mfaAuthenticated": "false",
                        "creationDate": "2016-04-13T17:51:37Z"
                    }
                },
                "invokedBy": "signin.amazonaws.com"
            },
            "eventTime": "2016-04-13T18:05:53Z",
```
Monitoring Trusted Advisor checks

AWS Trusted Advisor checks identify ways for you to reduce cost, increase performance, and improve security for your AWS account. You can use Amazon CloudWatch Events to monitor the status of Trusted Advisor checks. You can then use Amazon CloudWatch to create alarms on Trusted Advisor metrics. These alarms notify you when the status changes for a Trusted Advisor check, such as an updated resource or a service quota that is reached.

For example, Trusted Advisor provides the Amazon S3 Bucket Permissions check. This check identifies if you have buckets that have open access permissions or allow access to any authenticated AWS user. If a bucket permission changes, the status changes for the Trusted Advisor check. CloudWatch Events detects this event and then sends you a notification so that you can take action.

Topics

- Monitoring Trusted Advisor check results with Amazon CloudWatch Events (p. 63)
- Creating Trusted Advisor alarms using CloudWatch (p. 64)

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

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 procedure described here to create a CloudWatch alarm for Trusted Advisor. Before you create alarms for Trusted Advisor metrics, 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**

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**.

**Compliance validation for AWS Support**

Third-party auditors assess the security and compliance of AWS Support as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see [AWS services in scope by compliance program](https://aws.amazon.com/about-aws/whats-new/compliance-resources/). For general information, see [AWS compliance programs](https://aws.amazon.com/about-aws/whats-new/compliance-resources/).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading reports in AWS Artifact](https://aws.amazon.com/about-aws/whats-new/compliance-resources/).

Your compliance responsibility when using AWS Support is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- **Security and compliance quick start guides** – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- **Architecting for HIPAA security and compliance whitepaper** – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- **AWS compliance resources** – This collection of workbooks and guides might apply to your industry and location.
Resilience in AWS Support

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS global infrastructure.

Infrastructure security in AWS Support

As a managed service, AWS Support is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of security processes whitepaper.

You use AWS published API calls to access AWS Support through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Configuration and vulnerability analysis in AWS Support

For AWS Trusted Advisor, AWS handles basic security tasks such as guest operating system (OS) and database patching, firewall configuration, and disaster recovery.

Configuration and IT controls are a shared responsibility between AWS and you, our customer. For more information, see the AWS shared responsibility model.
Troubleshooting resources

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 AWS Support. The following table provides links to troubleshooting topics, arranged by service.

<table>
<thead>
<tr>
<th>Service</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Web Services</td>
<td>Troubleshooting AWS Signature Version 4 errors</td>
</tr>
<tr>
<td>Amazon AppStream</td>
<td>Troubleshoot Amazon AppStream</td>
</tr>
<tr>
<td>Amazon EC2 Auto Scaling</td>
<td>Troubleshooting Auto Scaling</td>
</tr>
<tr>
<td>AWS Certificate Manager (ACM)</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>AWS CloudFormation</td>
<td>Troubleshooting AWS CloudFormation</td>
</tr>
<tr>
<td>Amazon CloudFront</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>AWS CloudHSM</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>Amazon CloudSearch</td>
<td>Troubleshooting Amazon CloudSearch</td>
</tr>
<tr>
<td>AWS CodeDeploy</td>
<td>Troubleshooting AWS CodeDeploy</td>
</tr>
<tr>
<td>AWS Data Pipeline</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>AWS Direct Connect</td>
<td>Troubleshooting AWS Direct Connect</td>
</tr>
<tr>
<td>AWS Directory Service</td>
<td>Troubleshooting AWS Directory Service administration issues</td>
</tr>
<tr>
<td>Amazon DynamoDB</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>AWS Elastic Beanstalk</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>Amazon Elastic Compute Cloud (Amazon EC2)</td>
<td>Troubleshooting instances</td>
</tr>
<tr>
<td>Amazon Elastic Container Service (Amazon ECS)</td>
<td>Amazon ECS troubleshooting</td>
</tr>
<tr>
<td>Service</td>
<td>Link</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>Elastic Load Balancing</td>
<td>Troubleshoot your application load balancers</td>
</tr>
<tr>
<td>Amazon EMR (Amazon EMR)</td>
<td>Troubleshoot a cluster</td>
</tr>
<tr>
<td>Amazon ElastiCache for Memcached</td>
<td>Troubleshooting applications</td>
</tr>
<tr>
<td>Amazon ElastiCache for Redis</td>
<td>Troubleshooting applications</td>
</tr>
<tr>
<td>AWS Flow Framework</td>
<td>Troubleshooting and debugging tips</td>
</tr>
<tr>
<td>AWS GovCloud (US)</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>AWS Identity and Access Management (IAM)</td>
<td>Troubleshooting IAM</td>
</tr>
<tr>
<td>Kinesis</td>
<td>Troubleshooting Kinesis producers</td>
</tr>
<tr>
<td>AWS Lambda</td>
<td>Troubleshooting and monitoring AWS Lambda functions with CloudWatch</td>
</tr>
<tr>
<td>AWS OpsWorks</td>
<td>Debugging and troubleshooting guide</td>
</tr>
<tr>
<td>Amazon Redshift</td>
<td>Troubleshooting queries</td>
</tr>
<tr>
<td>Amazon Relational Database Service (Amazon RDS)</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>Amazon Route 53</td>
<td>Troubleshooting Amazon Route 53</td>
</tr>
<tr>
<td>Amazon Silk</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>Amazon Simple Email Service (Amazon SES)</td>
<td>Troubleshooting Amazon SES</td>
</tr>
<tr>
<td>Amazon Simple Storage Service (Amazon S3)</td>
<td>Troubleshooting CORS issues</td>
</tr>
<tr>
<td>Amazon Simple Workflow Service (Amazon SWF)</td>
<td>AWS flow framework for Java: Troubleshooting and debugging tips</td>
</tr>
<tr>
<td>AWS Storage Gateway</td>
<td>Troubleshooting your gateway</td>
</tr>
<tr>
<td>Amazon Virtual Private Cloud (Amazon VPC)</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>Amazon WorkMail</td>
<td>Troubleshooting the Amazon WorkMail web application</td>
</tr>
<tr>
<td>Amazon WorkSpaces</td>
<td>Troubleshooting Amazon WorkSpaces administration issues</td>
</tr>
<tr>
<td>Amazon WorkSpaces Application Manager (Amazon WAM)</td>
<td>Troubleshooting Amazon WAM application issues</td>
</tr>
</tbody>
</table>
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:** August 4, 2020

<table>
<thead>
<tr>
<th>Change</th>
<th>Description</th>
<th>Date changed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational view for AWS Trusted Advisor</td>
<td>You can now create reports for Trusted Advisor checks for accounts that are part of AWS Organizations. See <a href="https://docs.aws.amazon.com/AWSSecurity/latest/S3Guide/organizational-view-trusted-advisor.html">Organizational view for AWS Trusted Advisor</a> (p. 9).</td>
<td>July 17, 2020</td>
</tr>
<tr>
<td>Security and AWS Support</td>
<td>Added information about security considerations when using AWS Support.</td>
<td>January 10, 2020</td>
</tr>
<tr>
<td>Using Trusted Advisor as a web service</td>
<td>Added updated instructions to refresh Trusted Advisor data after getting list of Trusted Advisor checks.</td>
<td>November 1, 2018</td>
</tr>
<tr>
<td>Using Service-Linked Roles</td>
<td>Added new section.</td>
<td>July 11, 2018</td>
</tr>
<tr>
<td>Getting Started: Troubleshooting</td>
<td>Added troubleshooting links for Route 53 and AWS Certificate Manager.</td>
<td>September 1, 2017</td>
</tr>
<tr>
<td>Case Management Example: Creating a Case</td>
<td>Added a note about the <strong>CC</strong> box for users who have the Basic support plan.</td>
<td>August 1, 2017</td>
</tr>
<tr>
<td>Monitoring Trusted Advisor Check Results with CloudWatch Events</td>
<td>Added new section.</td>
<td>November 18, 2016</td>
</tr>
<tr>
<td>Case Management</td>
<td>Updated the names of case severity levels.</td>
<td>October 27, 2016</td>
</tr>
<tr>
<td>Logging AWS Support API Calls with AWS CloudTrail</td>
<td>Added new section.</td>
<td>April 21, 2016</td>
</tr>
<tr>
<td>Getting Started: Troubleshooting</td>
<td>Added more troubleshooting links.</td>
<td>May 19, 2015</td>
</tr>
<tr>
<td>Change</td>
<td>Description</td>
<td>Date changed</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Getting Started: Troubleshooting</td>
<td>Added more troubleshooting links.</td>
<td>November 18, 2014</td>
</tr>
<tr>
<td>Getting Started: Case Management</td>
<td>Updated to reflect AWS Service Catalog in the AWS Management Console.</td>
<td>October 30, 2014</td>
</tr>
<tr>
<td>Programming the Life of an AWS Support Case</td>
<td>Added information about new API elements for adding attachments to cases and for omitting case communications when retrieving case history.</td>
<td>July 16, 2014</td>
</tr>
<tr>
<td>Accessing AWS Support</td>
<td>Removed named support contacts as an access method.</td>
<td>May 28, 2014</td>
</tr>
<tr>
<td>Getting Started</td>
<td>Added the Getting Started section.</td>
<td>December 13, 2013</td>
</tr>
<tr>
<td>Initial publication</td>
<td>New AWS Support service released.</td>
<td>April 30, 2013</td>
</tr>
</tbody>
</table>
AWS glossary

For the latest AWS terminology, see the AWS glossary in the AWS General Reference.