User Guide

AWS Diagnostic Tools

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Diagnostic Tools: User Guide

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

What is AWS Diagnostic Tools?

AWS Diagnostic Tools is a service for AWS Partners in the Partner-Led Support program. Deployed on customer accounts managed by AWS Partners, Diagnostic Tools provides support engineers with a dashboard of over 50 context-aware tools that operate on their customer's AWS account. Using these tools, engineers supporting their customers can run tools to diagnose up to 22 AWS services running on their customer's AWS account.

The AWS Diagnostic Tools service provides a secure method for engineers in a partner organization to streamline the diagnostic process, enabling rapid access to necessary service metadata without accessing the service console on their customer's AWS account. The robust tagging feature enables partners to link tool executions from various tools to specific cases in their case management systems. This integration allows engineers to easily review all tool executions related to a particular case, facilitating efficient collaboration among teams distributed across multiple locations.

Most tools in the AWS Diagnostic Tools service provide a cross-Region view of each service on the account. This speeds up the support process for engineers, allowing them to move quickly while troubleshooting.

Tip

Learn more about the Partner-Led Support program.

Get started with AWS Diagnostic Tools

Topics

Overview

To access the AWS Diagnostic Tools service, your account must be managed by a Partner-Led Support (PLS) program partner.

> **Tip**
>
> Learn more about the Partner-Led Support program.

Your partners use a role with read-only IAM permissions that you create to run Diagnostic Tools, which perform read-only operations on your AWS account. This helps your partners quickly investigate issues with your AWS services or applications deployed on your AWS account.

> **Note**
>
> Diagnostic Tools is only visible on your account if your account is managed by an AWS Partner that is a part of the AWS Partner-Led Support program. If your account is managed by an AWS Partner under the Partner-Led Support program, and you don't see Diagnostic Tools, please contact your partner to investigate.

Enabling Diagnostic Tools on your account

Before you can enable the AWS Diagnostic Tools service on your account, you must meet the following prerequisites:

1. Your partner must be in the Partner-Led Support program.

2. Your partner should enlist your account as a managed account based on your support plan with your partner.
3. For each account managed by your partner, you must create the read-only IAM role with the permission policy attached to the trust policy that can execute Diagnostic Tools on your account.
4. Your partner must use the IAM role you create to federate into the account to use Diagnostic Tools.

Follow these instructions to create the read-only IAM role for Diagnostic Tools on your account:

Step 1: Create a permission policy

Create a permission policy with the correct API permissions

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. Choose **Policy**, then choose **JSON**.
3. Copy and paste the following policy.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "acm-pca:ListCertificateAuthorities",
        "acm-pca:describeCertificateAuthority",
        "acm-pca:describeCertificateAuthorityAuditReport",
        "acm-pca:getCertificateAuthorityCertificate",
        "acm-pca:getCertificateAuthorityCsr",
        "acm-pca:listTags",
        "acm:describeCertificate",
        "acm:getCertificate",
        "acm:listCertificates",
        "acm:listTagsForCertificate",
        "cloudfront:listDistributionsByWebACLId",
        "cloudtrail:describeTrails",
        "cloudtrail:getEventSelectors",
```

```
            "cloudtrail:lookupEvents",
            "cloudwatch:getMetricData",
            "cloudwatch:listDashboards",
            "cloudwatch:getDashboard",
            "cloudwatch:listMetrics",
            "codepipeline:getPipeline",
            "codepipeline:getPipelineState",
            "codepipeline:listActionTypes",
            "codepipeline:listPipelineExecutions",
            "codepipeline:listPipelines",
            "ec2:describeCapacityReservations",
            "ec2:describeByoipCidrs",
            "ec2:describeDhcpOptions",
            "ec2:describeNatGateways",
            "ec2:describeNetworkAcls",
            "ec2:describeNetworkInterfaces",
            "ec2:describePublicIpv4Pools",
            "ec2:describeRouteTables",
            "ec2:describeSecurityGroups",
            "ec2:describeSpotFleetRequests",
            "ec2:describeSpotInstanceRequests",
            "ec2:describeSubnets",
            "ec2:describeVpcs",
            "elasticfilesystem:describeAccessPoints",
            "elasticfilesystem:describeFileSystemPolicy",
            "elasticfilesystem:describeFileSystems",
            "elasticfilesystem:describeLifecycleConfiguration",
            "elasticfilesystem:describeMountTargets",
            "elasticfilesystem:listTagsForResource",
            "elasticloadbalancing:describeListeners",
            "elasticloadbalancing:describeLoadBalancers",
            "elasticloadbalancing:describeTags",
            "elasticloadbalancing:describeTargetGroups",
            "elasticloadbalancing:describeTargetHealth",
            "events:describeRule",
            "events:listApiDestinations",
            "events:listConnections",
            "events:listEventBuses",
            "events:listEventSources",
            "events:listRules",
            "events:listTargetsByRule",
            "guardduty:getFindings",
            "guardduty:listDetectors",
            "guardduty:listFindings",
```

```
            "guardduty:listIPSets",
            "guardduty:listThreatIntelSets",
            "iam:getAccessKeyLastUsed",
            "iam:getGroupPolicy",
            "iam:getPolicy",
            "iam:getPolicyVersion",
            "iam:getRole",
            "iam:getRolePolicy",
            "iam:getServerCertificate",
            "iam:getUser",
            "iam:getUserPolicy",
            "iam:listAccessKeys",
            "iam:listAttachedGroupPolicies",
            "iam:listAttachedRolePolicies",
            "iam:listAttachedUserPolicies",
            "iam:listGroupPolicies",
            "iam:listGroupsForUser",
            "iam:listInstanceProfiles",
            "iam:listMFADevices",
            "iam:listPolicies",
            "iam:listPolicyVersions",
            "iam:listRolePolicies",
            "iam:listRoles",
            "iam:listSSHPublicKeys",
            "iam:listServerCertificates",
            "iam:listUserPolicies",
            "iam:listUsers",
            "iam:listVirtualMFADevices",
            "lambda:getAccountSettings",
            "lambda:listEventSourceMappings",
            "lambda:listFunctions",
            "lambda:listLayers",
            "lambda:getFunction",
            "lambda:getPolicy",
            "lambda:listAliases",
            "lambda:listProvisionedConcurrencyConfigs",
            "lambda:listVersionsByFunction",
            "logs:describeExportTasks",
            "logs:describeLogGroups",
            "logs:describeLogStreams",
            "logs:describeMetricFilters",
            "logs:describeSubscriptionFilters",
            "medialive:listChannels",
            "medialive:listInputSecurityGroups",
```

```
            "medialive:listInputs",
            "mobiletargeting:getAdmChannel",
            "mobiletargeting:getApnsChannel",
            "mobiletargeting:getApnsSandboxChannel",
            "mobiletargeting:getApnsVoipChannel",
            "mobiletargeting:getApnsVoipSandboxChannel",
            "mobiletargeting:getApplicationSettings",
            "mobiletargeting:getApps",
            "mobiletargeting:getBaiduChannel",
            "mobiletargeting:getCampaign",
            "mobiletargeting:getCampaignActivities",
            "mobiletargeting:getCampaignVersions",
            "mobiletargeting:getCampaigns",
            "mobiletargeting:getEmailChannel",
            "mobiletargeting:getEventStream",
            "mobiletargeting:getExportJobs",
            "mobiletargeting:getGcmChannel",
            "mobiletargeting:getImportJobs",
            "mobiletargeting:getJourney",
            "mobiletargeting:getJourneyExecutionActivityMetrics",
            "mobiletargeting:getJourneyExecutionMetrics",
            "mobiletargeting:getJourneyRunExecutionActivityMetrics",
            "mobiletargeting:getJourneyRuns",
            "mobiletargeting:getSegment",
            "mobiletargeting:getSegmentImportJobs",
            "mobiletargeting:getSegmentVersions",
            "mobiletargeting:getSegments",
            "mobiletargeting:getSmsChannel",
            "mobiletargeting:listJourneys",
            "pipes:listPipes",
            "polly:describeVoices",
            "polly:listLexicons",
            "rds:describeDBClusterParameterGroups",
            "rds:describeDBClusterParameters",
            "rds:describeDBClusterSnapshots",
            "rds:describeDBClusters",
            "rds:describeDBInstances",
            "rds:describeDBParameterGroups",
            "rds:describeDBParameters",
            "rds:describeDBSecurityGroups",
            "rds:describeDBSnapshots",
            "rds:describeDBSubnetGroups",
            "rds:describeEvents",
            "rds:describePendingMaintenanceActions",
```

```
            "rds:listTagsForResource",
            "redshift:describeClusterParameterGroups",
            "redshift:describeClusterParameters",
            "redshift:describeClusterSnapshots",
            "redshift:describeClusterSubnetGroups",
            "redshift:describeClusters",
            "redshift:describeEventSubscriptions",
            "redshift:describeEvents",
            "redshift:describeLoggingStatus",
            "redshift:describeReservedNodes",
            "redshift:describeResize",
            "route53domains:getDomainDetail",
            "route53domains:getOperationDetail",
            "route53domains:listDomains",
            "route53domains:listOperations",
            "scheduler:listScheduleGroups",
            "scheduler:listSchedules",
            "servicequotas:listAWSDefaultServiceQuotas",
            "servicequotas:listServiceQuotas",
            "ssm:describeActivations",
            "ssm:describeAutomationExecutions",
            "ssm:describeInstanceInformation",
            "ssm:describeMaintenanceWindows",
            "ssm:describeParameters",
            "ssm:describePatchBaselines",
            "ssm:describePatchGroups",
            "ssm:listDocuments",
            "swf:describeActivityType",
            "swf:describeDomain",
            "swf:describeWorkflowExecution",
            "swf:describeWorkflowType",
            "swf:getWorkflowExecutionHistory",
            "swf:listActivityTypes",
            "swf:listClosedWorkflowExecutions",
            "swf:listDomains",
            "swf:listOpenWorkflowExecutions",
            "swf:listWorkflowTypes",
            "waf-regional:getWebACL",
            "waf-regional:listResourcesForWebACL",
            "waf-regional:listWebACLs",
            "waf:getWebACL",
            "waf:listWebACLs"
        ],
         "Resource": "*"
```

```
    }
  ]
}
```

4. Give the policy a name. Note this name. You will need it in a later step.
5. Add a description.

Step 2: Create a trust policy and add permissions

1. In the IAM dashboard, choose **Roles** on the left panel.
2. Choose **Create role**.
3. Select the **Custom trust policy** option.
4. Copy and paste the following into the **Custom trust policy** panel.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": [
                "ts.amazonaws.com"
            ]
        },
        "Action": "sts:AssumeRole"
    }]
}
```

5. Choose **Next**. Select **Add permissions**, then select **Attach policies**.
6. Search for the permission policy you created and select it.
7. Choose **Next**, then enter a name and description.

> **Note**
>
> You share this permission with your partner. Enter a name you can use to recognize this role.

8. Create the IAM role. Share the name of this IAM role with your partner.

Partner training for Diagnostic Tools

As a part of your partner's participation in the Partner-Led Support program, we provide training on how to use Diagnostic Tools and when to use them.

> **Tip**
>
> Your partners can access their training content at the AWS Skill Builder site.

Security in AWS Diagnostic Tools

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are 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 Diagnostic Tools, 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 Diagnostic Tools. The following topics show you how to configure Diagnostic Tools to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Diagnostic Tools resources.

Topics

Identity and Access Management for AWS Diagnostic Tools

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 Diagnostic Tools resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting AWS Diagnostic Tools identity and access)
- **Service administrator** - determine user access and submit permission requests (see How AWS Diagnostic Tools works with IAM)
- **IAM administrator** - write policies to manage access (see Identity-based policy examples for AWS Diagnostic Tools)

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be authenticated as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users to use federation with an identity provider to access AWS services using temporary credentials.

A *federated identity* is a user from your enterprise directory, web identity provider, or Directory Service that accesses AWS services using credentials from an identity source. Federated identities assume roles that provide temporary credentials.

For centralized access management, we recommend AWS IAM Identity Center. For more information, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

Identity-based policies can be *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, see Choose between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples include IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. You must specify a principal in a resource-based policy.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. For more information, see Session policies in the *IAM 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*.

How AWS Diagnostic Tools works with IAM

Before you use IAM to manage access to Diagnostic Tools, learn what IAM features are available to use with Diagnostic Tools.

IAM features you can use with AWS Diagnostic Tools

IAM feature	Diagnostic Tools support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No

IAM feature	Diagnostic Tools support
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Forward access sessions (FAS)	Partial
Service roles	No
Service-linked roles	No

To get a high-level view of how Diagnostic Tools and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Diagnostic Tools

Supports identity-based policies: Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Define custom IAM permissions with customer managed policies in the *IAM User Guide*.

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. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Diagnostic Tools

To view examples of Diagnostic Tools identity-based policies, see Identity-based policy examples for AWS Diagnostic Tools.

Resource-based policies within Diagnostic Tools

Supports resource-based policies: No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that

support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Diagnostic Tools

Supports policy actions: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Diagnostic Tools actions, see Actions Defined by Diagnostic Tools in the *Service Authorization Reference*.

Policy actions in Diagnostic Tools use the following prefix before the action:

```
ts
```

To specify multiple actions in a single statement, separate them with commas.

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

To view examples of Diagnostic Tools identity-based policies, see Identity-based policy examples for AWS Diagnostic Tools.

Policy resources for Diagnostic Tools

Supports policy resources: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Diagnostic Tools resource types and their ARNs, see Resources Defined by Diagnostic Tools in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by Diagnostic Tools .

To view examples of Diagnostic Tools identity-based policies, see Identity-based policy examples for AWS Diagnostic Tools.

Policy condition keys for Diagnostic Tools

Supports service-specific policy condition keys: Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Diagnostic Tools condition keys, see Condition Keys for Diagnostic Tools in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions Defined by Diagnostic Tools .

To view examples of Diagnostic Tools identity-based policies, see Identity-based policy examples for AWS Diagnostic Tools.

ACLs in Diagnostic Tools

Supports ACLs: No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with Diagnostic Tools

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see Define permissions with ABAC authorization in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with Diagnostic Tools

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Forward access sessions for Diagnostic Tools

Supports forward access sessions (FAS): Partial

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service roles for Diagnostic Tools

Supports service roles: No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break Diagnostic Tools functionality. Edit service roles only when Diagnostic Tools provides guidance to do so.

Service-linked roles for Diagnostic Tools

Supports service-linked roles: No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a Yes in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for AWS Diagnostic Tools

By default, users and roles don't have permission to create or modify Diagnostic Tools resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Create IAM policies (console) in the *IAM User Guide*.

For details about actions and resource types defined by Diagnostic Tools, including the format of the ARNs for each of the resource types, see Actions, Resources, and Condition Keys for Diagnostic Tools in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Diagnostic Tools resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies

adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.

- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Diagnostic Tools console

To access the Diagnostic Tools console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Diagnostic Tools 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 (users or roles) with that policy.

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 they're trying to perform.

To ensure that users and roles can still use the Diagnostic Tools console, also attach the Diagnostic Tools `ConsoleAccess` or `ReadOnly` AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allowing all operations on the Diagnostic Tools service

In this example, you want to grant an IAM user in your AWS account access to run all operations on the Diagnostic Tools service on your account. You also want to allow the user to list executions, tools and tags for resources.

Allowing `StartExecution` for a specific tool

In this example, you want to grant an IAM user in your AWS account access to run a specific tools using a specific passrole. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

Allowing `StartExecution` using a specific `IAM user`

In this example, you want to grant an IAM user in your AWS account access to run all tools using a specific passrole. 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": [
            {
               "Effect": "Allow",
               "Action": ["ts:*"],
               "Resource": "*"
            },
            {
               "Effect": "Allow",
               "Action": ["iam:PassRole"],
               "Resource": "arn:aws:iam::000000000000:role/
DiagnosticToolSampleRole"
            }
         ]
   }
```

Troubleshooting AWS Diagnostic Tools identity and access

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

Topics

- I am not authorized to perform an action in Diagnostic Tools
- I am not authorized to perform iam:PassRole
- I want to allow people outside of my AWS account to access my Diagnostic Tools resources

I am not authorized to perform an action in Diagnostic Tools

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `ts:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ts:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `ts:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Diagnostic Tools.

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 Diagnostic Tools. However, the action requires the service to have permissions that are 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's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my Diagnostic Tools 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 Diagnostic Tools supports these features, see How AWS Diagnostic Tools works with IAM.
- 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*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Compliance validation for AWS Diagnostic Tools

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Compliance validation

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html.

Your compliance responsibility when using AWS services 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 baseline environments on AWS that are security and compliance focused. Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

When using Diagnostic Tools to diagnose AWS services in your account, you can select the Region to store the tool output. To meet data sovereignty requirements for a jurisdiction, specify where to store the output. The output saved in the destination Region will not replicate to other Regions. This facilitates data sovereignty for regulations like General Data Protection Regulation (GDPR) and regional requirements.

Note

Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.
AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
Evaluating Resources with Rules in the AWS Config Developer Guide – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
AWS Security Hub CSPM – This AWS service provides a comprehensive view of your security state within AWS. Security Hub CSPM uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.

AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Data protection in AWS Diagnostic Tools

The AWS shared responsibility model applies to data protection in Diagnostic Tools. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, 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. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- 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 sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Diagnostic Tools or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used

for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Data encryption

AWS Diagnostic Tools data is encrypted in transit and at rest. When you submit data to Diagnostic Tools, it encrypts the data as it receives and stores it. When you retrieve data from Diagnostic Tools, it transmits the data to you by using current security protocols. Additionally, when you run a tool, the output of the Diagnostic Tools service encrypts the tool output generated by the service.

Encryption at rest

Diagnostic Tools encrypts all the data that it stores for you. This includes configuration data, user and endpoint data, analytics data, and any data that you add or import into Diagnostic Tools. To encrypt your data, Diagnostic Tools uses internal AWS Key Management Service (AWS KMS) keys that the service owns and maintains. We rotate these keys on a regular basis. For information about AWS KMS, see the AWS Key Management Service Developer Guide.

Encryption in transit

Diagnostic Tools uses HTTPS and Transport Layer Security (TLS) 1.2 or later to communicate with your clients and applications. To communicate with other AWS services, Diagnostic Tools uses HTTPS and TLS 1.2. In addition, when you create and manage Diagnostic Tools resources by using the console, an AWS SDK, or the AWS Command Line Interface, all communications are secured using HTTPS and TLS 1.2.

Key management

Diagnostic Tools uses HTTPS and Transport Layer Security (TLS) 1.2 or later to communicate with your clients and applications. To communicate with other AWS services, Diagnostic Tools uses HTTPS and TLS 1.2. In addition, when you create and manage Diagnostic Tools resources by using the console, an AWS SDK, or the AWS Command Line Interface, all communications are secured using HTTPS and TLS 1.2.

Internetwork privacy

You can use Amazon Virtual Private Cloud (Amazon VPC) to create boundaries between resources in your managed nodes and control traffic between them, your on-premises network, and the internet. For details, see Create VPC endpoints..

For more information about Amazon Virtual Private Cloud security, see Internetwork traffic privacy in Amazon VPC in the *Amazon VPC User Guide*.

Resilience in AWS Diagnostic Tools

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.

Logging and monitoring in AWS Diagnostic Tools

AWS Diagnostic Tools provides logging and monitoring capabilities to help users track and analyze the actions taken within the service. This includes integrating with AWS CloudTrail to log API calls, and using Amazon CloudWatch to collect and analyze service metrics. The service also retains tool output data for up to 30 days.

CloudTrail

AWS Diagnostic Tools integrates with AWS CloudTrail, which is a service that provides a record of actions that were taken in Diagnostic Tools by a user, a role, or another AWS service. This includes actions from the Diagnostic Tools console and programmatic calls to Diagnostic Tools API operations. By using the information collected by CloudTrail, you can determine which requests were made to Diagnostic Tools. For each request, you can identify when it was made, the IP address from which it was made, who made it, and additional details. For more information, see Logging Diagnostic Tools API calls with AWS CloudTrail.

CloudWatch

You can use Amazon CloudWatch to collect, view, and analyze several Service Quotas metrics related to the Diagnostic Tools service running on your account. You can use Amazon CloudWatch to create alarms that notify you if the value for Service Quotas metrics captured by CloudWatch meet or exceed certain conditions or is within or exceeds a threshold that you define. If you create an alarm, you can configure CloudWatch to send a notification to an Amazon Simple Notification Service (Amazon SNS) topic that you specify.

Tool output data retention

Tool output generated by the Diagnostic Tools service is stored in the Output destination you select at the time of running a tool. All tool output is retained for up to 30 days since the tool was run and cannot be restored.

Identify and diagnose issues with AWS Diagnostic Tools

The AWS Diagnostic Tools service provides a range of diagnostic tools you can use with a variety of AWS services on your partner-managed account, allowing you to discover issues with your applications or AWS services. Each tool operates on a list of AWS Regions selected when the tool is run. All invocations of each tool are logged, are easily accessible for review, and the output from each invocation can be directed to one of several different regions. The tools can be invoked from the AWS Management Console, with API access available in order to support tools, automation, and integration.

> **Note**
>
> Diagnostic Tools is only visible on your account if your account is managed by an AWS Partner that is a part of the AWS Partner-Led Support program. If your account is managed by an AWS Partner under the Partner-Led Support program, and you don't see Diagnostic Tools, please contact your partner to investigate.

Contents

Tools to diagnose application integrations

The tools used to diagnose application integrations in the AWS Diagnostic Tools service cover Amazon Pinpoint, Amazon Polly, and Amazon Simple Workflow Service. With Diagnostic Tools, AWS Partners can monitor, troubleshoot, and optimize their customer's AWS applications.

Amazon Pinpoint tools

Amazon Pinpoint diagnostic tools are employed to monitor user engagement and interactions within applications. This encompasses tracking user behavior, sending targeted messages, and evaluating the effectiveness of campaigns.

These tools aid in troubleshooting issues related to user engagement and message delivery, including:

- Diagnosing message delivery problems such as bounces or complaints.
- Investigating low user engagement by analyzing user interaction data.
- Optimizing message campaigns based on open rates, click-through rates, and conversion rates.

Amazon Pinpoint tools

- Amazon Pinpoint Campaign Details
- Amazon Pinpoint Segment Details
- Amazon Pinpoint Journey Details
- Amazon Pinpoint Journey Run Details
- Amazon Pinpoint Dashboard

Amazon Polly tools

The Amazon Polly tool helps partners convert text into speech and customize the speech output.

This tool helps partners troubleshoot text-to-speech conversion issues that may originate from Amazon Polly including:

- Identifying errors or mispronunciations in the generated speech.
- Analyzing usage patterns to ensure efficient utilization of Polly resources.
- Optimizing voice selection and speech markup to enhance speech quality.

Amazon Polly tools

- Amazon Polly

Amazon EventBridge tools

Amazon EventBridge Bridge is a serverless event bus service that enables software applications to communicate with each other using events. Offered by AWS, this service is designed to simplify the architecture of event-driven applications by providing a robust and scalable infrastructure for routing events between software components. Amazon EventBridge tools help troubleshoot EventBridge Rules:

- Quickly identify and list EventBridge Rules to confirm the presence and proper configuration of the correct rules for your events. This proves particularly valuable in complex systems with many Rules.
- Inspect event patterns within EventBridge Rules to pinpoint precisely which events a rule is monitoring. This helps diagnose issues, especially if a service isn't reacting as expected, potentially due to a mismatch in the event pattern and emitted events.
- Verify the targets of EventBridge Rules to confirm that events are correctly directed to the intended AWS service or resource. This step is crucial as misconfigured targets often disrupt event-driven architectures.
- Understand the details of event data transformation within EventBridge Rules for effective debugging. Missteps in transformation logic are a common source of unexpected application behavior.
- Ensure the proper permissions are associated with EventBridge Rules. Correct permissions are essential for the rule's operation; lacking them can result in untriggered rules or non-responsive targets.

Amazon EventBridge tools

- EventBridge Rule Details
- EventBridge Lookup

Amazon Simple Workflow Service tools

Amazon Simple Workflow Service tools help diagnose issues related to workflow orchestration and task execution, including:

- Detecting and resolving workflow execution failures or timeouts.
- Monitoring the progress of workflow executions and identifying bottlenecks.

- Debugging task failures by examining task history and input/output data.

Amazon Simple Workflow Service (Amazon SWF) tools

- Amazon Simple Workflow Service Activities
- Amazon Simple Workflow Service Dashboard
- Amazon Simple Workflow Service Domains
- Amazon Simple Workflow Service Execution Details
- Amazon Simple Workflow Service Execution History
- Amazon Simple Workflow Service List Executions
- Amazon Simple Workflow Service Types Lookup

Benefits for AWS Partners

By using AWS Diagnostic Tools your partners can handle common application integration related troubleshooting scenarios that may include:

- *Message delivery issues:* Troubleshooting message delivery problems in Amazon Pinpoint, investigating delivery logs, and ensuring message accuracy.
- *User engagement optimization:* Analyzing user engagement metrics in Amazon Pinpoint to improve user interactions and campaign effectiveness.
- *Voice quality and pronunciation:* Diagnosing voice quality issues and mispronunciations in Amazon Polly.
- *Resource utilization:* Monitoring resource utilization in Amazon Polly and Amazon SWF to optimize resource allocation.
- *Workflow execution failures:* Resolving workflow execution failures and timeouts in Amazon SWF.
- *Performance optimization:* Analyzing Amazon SWF workflows for performance bottlenecks and delays.
- *Task retries:* Handling task retries in Amazon SWF to ensure smooth workflow execution.

Tools for managing and optimizing AWS compute infrastructure and applications

The compute tools in the AWS Diagnostic Tools service are used for monitoring, troubleshooting, and optimizing AWS infrastructure and applications on your AWS accounts.

Amazon EC2 diagnostic tools

Amazon Elastic Compute Cloud (Amazon EC2) diagnostic tools provide you insights into your Amazon EC2 instances' performance, status, and resource utilization. You can use these tools to identify and resolve issues such as high CPU or memory usage, network connectivity problems, instance status checks failures, and disk space limitations.

Amazon EC2 tools

- Amazon EC2 Spot Fleet Requests Lookup
- Amazon EC2 Spot Instance Requests Lookup
- Amazon EC2 Systems Manager
- Amazon EC2 Capacity Reservation

AWS Lambda diagnostic tools

Lambda diagnostic tools are designed to enhance the visibility and manageability of your AWS Lambda environment, contributing to a more efficient and reliable serverless architecture.

Lambda tools

- Lambda Functions Lookup - provides AWS users with a comprehensive overview of all Lambda functions within their AWS account. This tool is designed to streamline management and diagnostic processes by offering a quick snapshot of the Lambda functions on an account.
- Lambda Function Details - provides an in-depth view of individual AWS Lambda functions, presenting a detailed configuration and operational overview. This tool is essential for troubleshooting, configuration verification, and ensuring optimal performance of your Lambda functions.

Application Load Balancer tools

Application Load Balancer (ALB) diagnostic tools help you monitor and manage your Application Load Balancers, including routing rules, target group configurations, and listener settings. With these tools, you can diagnose routing issues, identify misconfigured listeners or target groups, and understand whether your load balancer settings need to be optimized for improved traffic distribution.

Application Load Balancer tools

- Application Load Balancer Target Group Details
- Application Load Balancer Target on Target Groups Lookup
- Application Load Balancer List Listeners

AWS Elemental Live tools

AWS Elemental Live tools are designed for media processing and streaming workflows. These provide insights into the health and performance of AWS Elemental MediaLive, helping troubleshoot issues related to video encoding, streaming latency, packaging, and content delivery.

AWS Elemental Live tools

- AWS Elemental MediaLive Lookup

Benefits for partners

Using these sets of tools, your partners are equipped to handle common compute related troubleshooting scenarios that may include the following:

- *Performance optimization:* Identify bottlenecks, resource overutilization, or underutilization in Amazon EC2 instances and Lambda functions. Adjust resource allocation or configurations for better performance.
- *Error analysis:* Diagnose errors or failures in Lambda functions, Application Load Balancer routing, or event-driven workflows. Investigate error logs and metrics to pinpoint the root cause.
- *Scaling challenges:* Monitor resource scaling in Amazon EC2 instances and Lambda functions to ensure they can handle varying workloads effectively. Adjust auto-scaling policies as needed.

- *Security and access control:* Ensure that IAM roles and permissions are correctly configured for Lambda functions and other AWS resources. Detect and remediate security vulnerabilities.
- *Latency and load balancing:* Analyze Application Load Balancer metrics to detect latency issues or uneven traffic distribution. Adjust target group settings and routing rules to optimize load balancing.
- *Media processing and streaming issues:* Troubleshoot media encoding failures, streaming latency, or content delivery problems with MediaLive tools. Optimize video encoding settings and CDN configurations.
- *Event-driven workflow debugging:* Identify issues in event-driven architectures using Amazon EventBridge. Check rule configurations, target configurations, and event source integrations to ensure events are handled correctly.

Tools for managing and optimizing databases

Databases tools in AWS Diagnostic Tools assist in diagnosing, monitoring, and optimizing Amazon Redshift and Amazon RDS database environments in an AWS account. These tools are designed to provide insights into various aspects of database clusters, instances, parameters, and performance metrics, allowing users to proactively address issues and monitor the operation of their database workloads.

Amazon Redshift tools

The suite of diagnostic tools for Amazon Redshift offers a comprehensive set of functionalities for diagnosing, monitoring, and optimizing Amazon Redshift clusters. These tools provide insights into cluster status, node configuration, event tracking, log management, parameter settings, and more. Users can utilize these tools to proactively identify issues within their Amazon Redshift clusters, monitor cluster health, adjust parameter configurations, manage event subscriptions, and optimize resource utilization. Whether it's tracking events, managing snapshots, or resizing clusters, the Amazon Redshift tools help users maintain efficiency and reliability of their Amazon Redshift database workloads in their AWS account.

Amazon Redshift tools

- Amazon Redshift Clusters Lookup
- Amazon Redshift Describe Events
- Amazon Redshift List Logs

- Amazon Redshift List Parameter Groups
- Amazon Redshift Parameter Group Details
- Amazon Redshift Reserved Nodes
- Amazon Redshift Resize Details
- Amazon Redshift Snapshots Lookup
- Amazon Redshift Subnet Groups Lookup
- Amazon Redshift Subscriptions Lookup

Amazon RDS tools

The suite of diagnostic tools for Amazon Relational Database Service (Amazon RDS) offers a comprehensive set of functionalities for diagnosing, monitoring, and optimizing Amazon RDS clusters. These tools provide insights into cluster status, node configuration, event tracking, log management, parameter settings, and more. Users can utilize these tools to proactively identify issues within their Amazon Redshift clusters, monitor cluster health, adjust parameter configurations, manage event subscriptions, and optimize resource utilization. Whether it's tracking events, managing snapshots, or resizing clusters, the Amazon Redshift tools help users maintain efficiency and reliability of their Amazon Redshift database workloads in their AWS account.

Amazon RDS tools

- Amazon RDS Cluster Details
- Amazon RDS Cluster Parameters
- Amazon RDS Clusters With Metrics
- Amazon RDS Dashboard
- Amazon RDS DB Parameter Details
- Amazon RDS Instance Details
- Amazon RDS Instances with Metrics
- Amazon RDS Maintenance Details
- Amazon RDS Simple Charts

Tools for diagnosing DevOps and deployment services

By using the DevOps and deployment tools in AWS Diagnostic Tools, users can diagnose and troubleshoot issues in their cloud-based development environments and deployments. Covering CloudWatch, CodePipeline, and Amazon Elastic File System (EFS) services, these tools provide valuable insights into AWS resource configurations, performance, and health. Whether you need to examine CloudWatch dashboards, investigate log groups, review CodePipeline setups, or inspect Amazon Elastic File System resources, these tools are your go-to resources for monitoring, diagnosing, and optimizing your AWS development and deployments.

CloudWatch tools

The CloudWatch diagnostic tools offer the ability to monitor, diagnose, and optimize cloud-based environments. With these tools, users can gain insights into CloudWatch dashboards, monitor metric data, and investigate log groups effectively.

Amazon CloudWatch tools

- CloudWatch Dashboard Details
- CloudWatch Dashboards Lookup
- CloudWatch LogGroup Lookup

Additional DevOps tools

The **Amazon EFS Lookup tool** is a diagnostic, read-only utility that provides insights into Amazon EFS resources. It offers detailed information about the current FileSystem, MountTargetSecurityGroups, MountTargets, and associated CloudWatch metrics for the selected AWS account and AWS Region. With this tool, users can diagnose and monitor their Amazon EFS configurations.

The **AWS CodePipeline Investigator tool** provides a summary of CodePipeline configurations within an AWS account. Users can view the status, configuration details, and other essential information related to pipelines utilized in continuous deployment release management. With the Amazon EFS Lookup tool, users can also troubleshooting pipeline errors, deployment issues, and permission-related problems.

Tools for diagnosing networking services

The diagnostic networking tools can help users effectively manage and troubleshoot various networking configurations, including Bring Your Own IP (BYOIP) in Amazon EC2, Amazon Route 53 domain actions, load balancer responses, and Amazon VPC security groups.

BYOIP in EC2 tool

By using the Bring your Own IP in EC2 (BYOIP) tool, users can effectively manage and troubleshoot Bring Your Own IP (BYOIP) configurations in Amazon EC2. BYOIP enables organizations to bring their own IPv4 address space into AWS and is used during integration and optimization of BYOIP resources. The tool enables CIDR Range investigation, helping investigate BYOIP configurations by specifying CIDR ranges and verifying their mapping to AWS resources for proper utilization. Meanwhile, the IPv4 Pool analysis provides insights into the utilization, health, and allocation of BYOIP IPv4 pools, ensuring efficient resource utilization.

Route 53 tool

Amazon Route 53 tool is designed to assist users in investigating domain actions and changes within their Route 53 domain configurations. It provides insights into both active and pending domain actions, as well as recent domain-related activities.

This tool helps your partners investigate their customer's Route 53 managed domains as follows:

- *Active Domain Actions:* Easily access a list of ongoing domain operations, including registrations, transfers, renewals, and DNS updates.
- *Pending Domain Actions:* Monitor pending tasks like transfers and renewals. Ensure they progress smoothly by checking statuses and expected completion dates.
- *Recent Domain Activities:* Review a historical log of domain-related events, from updates to DNS changes. Useful for auditing and tracking.
- *Troubleshooting and Monitoring:* Empower yourself to resolve issues, track task progress, and maintain domain accuracy and security. Detect unauthorized changes swiftly.

Load Balancer Responses

Overview The Application Load Balancer Responses tool provides basic information and operational insights into the Application, Network, and Classic load balancers within the AWS environment across selected Regions.

Key features:

The tool fetches and displays the following fields for each Application/Network Load Balancer and Classic Load Balancer:

- *FQDN (Fully Qualified Domain Name):* Displays the fully qualified domain name associated with the load balancer.
- *Type:* Identifies the type of load balancer (Application, Network, or Classic).
- *Scheme:* Indicates whether the load balancer is internet-facing or internal.
- *Addressing:* Shows the type of IP address used (static or dynamic).
- *VPC (Virtual Private Cloud):* Lists the VPC in which the load balancer is deployed.
- *AZs (Availability Zones):* Displays the Availability Zones where the load balancer is available.
- *Listeners:* Enumerates the listeners configured for the load balancer.
- *Target Groups:* (For Application and Network Load Balancers): Lists the target groups associated with the load balancer.
- *Creation Time:* Shows the timestamp when the load balancer was created.
- *Region:* Indicates the AWS Region where the load balancer is deployed.

VPC Security Groups Lookup

The Amazon Virtual Private Cloud (Amazon VPC) Security Groups Lookup tool simplifies the task of listing all VPC security groups within your AWS account. It offers a quick and efficient way to gather essential information about the security groups in use, aiding in the management and oversight of your VPC configurations. Whether you need to verify security settings or ensure compliance, this tool provides an essential asset for VPC security management.

Tools for maintaining security, compliance, and efficient performance

Security diagnostic tools available in the AWS Diagnostic Tools service equip AWS users with resources for maintaining security, compliance, and efficient performance in AWS environments. These tools offer certificate management, effective access policy control, and streamlined threat detection and response. AWS Partners in the Partner-Led Support program can leverage these resources to diagnose issues promptly, ensure resource integrity, and enhance their customer's AWS experience.

AWS Certificate Manager tools

The AWS Certificate Manager tools offer valuable resources for managing and troubleshooting SSL/TLS certificates in AWS. With tools like Audit Report Details, Authority Certificate Details, and Certificates List, users can ensure the security and compliance of their certificates by investigating audit reports, authority certificates, and certificate details. Additionally, Certificate Manager Signing Request allows users to review and confirm the content of certificate signing requests, aiding in troubleshooting issues related to certificate creation and management.

AWS Certificate Manager tools

- AWS Certificate Manager Audit Report Details
- AWS Certificate Manager Authority Certificate Details
- AWS Certificate Manager Authority Lookup
- AWS Certificate Manager Certificate Authority Details
- AWS Certificate Manager Certificate Details
- AWS Certificate Manager Certificates Lookup
- AWS Certificate Manager Signing Request Lookup

GuardDuty tool

The GuardDuty Findings tool helps with security monitoring and threat detection in AWS. It provides a centralized view of GuardDuty findings across AWS Regions, enabling users to swiftly identify and investigate security threats and vulnerabilities. With this tool, users can take prompt action to mitigate security risks, enhancing the overall security posture of their AWS environment.

Web ACL tool

The Classic Web Access Control Lists (Web ACL) tool allows users to view a list of all Classic Web ACLs within the AWS account across all AWS Regions. It provides an overview of ACL names, configurations, and associated resources, enabling users to quickly assess the status of their ACLs. Users can view individual Classic Web ACLs with this tool and gather insights into ACL rules, rule evaluations, associated resources, and rule actions. The tool facilitates improved security through proactive identification and resolution of security vulnerabilities in Classic Web ACL configurations, while also simplifying troubleshooting with ACL insights for expedited issue diagnosis and resolution.

IAM tools

The IAM tools offer comprehensive insights into AWS Identity and Access Management (IAM) configurations, user policies, and access controls in AWS. Access Keys Lookup allows users to review and secure access keys, while the IAM Dashboard offers an overall view of IAM resources. IAM Policy Roles Lookup helps diagnose and fine-tune IAM role policies, and IAM Policy Versions Lookup provides policy history information. IAM Server Certificates Lookup assists in managing server certificates, and IAM User Policies Lookup supports secure user access configurations. These tools help Partner-Led Support partners maintain robust IAM configurations and secure access to their customer's AWS resources.

AWS Identity and Access Management tools

- IAM Access Keys Lookup
- IAM Dashboard
- IAM Policy Roles Lookup
- IAM Policy Versions Lookup
- IAM Server Certificates Lookup
- IAM User Policies Lookup

Document history for AWS Diagnostic Tools

The following table describes the important changes to the documentation since the last release of AWS Diagnostic Tools. For notification about updates to this documentation, subscribe to the RSS feed.

- **Latest documentation update:** October 23, 2024

Change	Description	Date
Content optimization and correction of permission policy	Introduced optimizations to page layout and content. Corrected errors in the previous version of the permission policy on Get started with AWS Diagnostic Tools page	October 23, 2024
Updates to documentation	Documented launch of 12 additional tools, corrected broken links, and updated policy on Get started with AWS Diagnostic Tools page	February 2, 2024
Launch version of documentation	Launched documentation for AWS Diagnostic Tools service	November 27, 2023