Grant a user permissions to pass a role to an AWS service

Archived: 2026-04-06 00:16:16 UTC

To configure many AWS services, you must *pass* an IAM role to the service. This allows the service to assume the role later and perform actions on your behalf. For most services, you only have to pass the role to the service once during setup, and not every time that the service assumes the role. For example, assume that you have an application running on an Amazon EC2 instance. That application requires temporary credentials for authentication, and permissions to authorize the application to perform actions in AWS. When you set up the application, you must pass a role to Amazon EC2 to use with the instance that provides those credentials. You define the permissions for the applications running on the instance by attaching an IAM policy to the role. The application assumes the role every time it needs to perform the actions that are allowed by the role.

To pass a role (and its permissions) to an AWS service, a user must have permissions to *pass the role* to the service. This helps administrators ensure that only approved users can configure a service with a role that grants permissions. To allow a user to pass a role to an AWS service, you must grant the `PassRole` permission to the user's IAM user, role, or group.

Warning

- You can only use the `PassRole` permission to pass an IAM role to a service that shares the same AWS account. To pass a role in Account A to a service in Account B, you must first create an IAM role in Account B that can assume the role from Account A, and then the role in Account B can be passed to the service. For details, see Cross account resource access in IAM.
- Do not try to control who can pass a role by tagging the role and then using the `ResourceTag` condition key in a policy with the `iam:PassRole` action. This approach does not have reliable results.

When setting the `PassRole` permission, you should make sure that a user doesn't pass a role where the role has more permissions than you want the user to have. For example, Alice might not be allowed to perform any Amazon S3 actions. If Alice could pass a role to a service that allows Amazon S3 actions, the service could perform Amazon S3 actions on behalf of Alice when executing the job.

When you specify a service-linked role, you must also have permission to pass that role to the service. Some services automatically create a service-linked role in your account when you perform an action in that service. For example, Amazon EC2 Auto Scaling creates the `AWSServiceRoleForAutoScaling` service-linked role for you when you create an Auto Scaling group for the first time. If you try to specify the service-linked role when you create an Auto Scaling group and you don't have the `iam:PassRole` permission, you receive an error. If you don't explicitly specify the role, the `iam:PassRole` permission is not required, and the default is to use `AWSServiceRoleForAutoScaling` role for all operations that are performed on that group. To learn which services support service-linked roles, see AWS services that work with IAM. To learn which services automatically create a service-linked role when you perform an action in that service, choose the **Yes** link and view the service-linked role documentation for the service.

A user can pass a role ARN as a parameter in any API operation that uses the role to assign permissions to the service. The service then checks whether that user has the `iam:PassRole` permission. To limit the user to passing only approved roles, you can filter the `iam:PassRole` permission with the `Resources` element of the IAM policy statement.

You can use the `Condition` element in a JSON policy to test the value of keys included in the request context of all AWS requests. To learn more about using condition keys in a policy, see IAM JSON policy elements: Condition. The `iam:PassedToService` condition key can be used to specify the service principal of the service to which a role can be passed. To learn more about using the `iam:PassedToService` condition key in a policy, see iam:PassedToService.

Example 1

Suppose you want to grant a user the ability to pass any of an approved set of roles to the Amazon EC2 service upon launching an instance. You need three elements:

- An IAM *permissions policy* attached to the role that determines what the role can do. Scope permissions to only the actions that the role must perform, and to only the resources that the role needs for those actions. You can use an AWS managed or customer-created IAM permissions policy.

```
  {
    "Version":"2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [ " A list of the permissions the role is allowed to use " ],
        "Resource": [ " A list of the resources the role is allowed to access " ]
    }
}
```

- A *trust policy* for the role that allows the service to assume the role. For example, you could attach the following trust policy to the role with the `UpdateAssumeRolePolicy` action. This trust policy allows Amazon EC2 to use the role and the permissions attached to the role.

```
  {
    "Version":"2012-10-17",
    "Statement": {
        "Sid": "TrustPolicyStatementThatAllowsEC2ServiceToAssumeTheAttachedRole",
        "Effect": "Allow",
        "Principal": { "Service": "ec2.amazonaws.com" },
       "Action": "sts:AssumeRole"
    }
}
```

- An IAM *permissions policy* attached to the IAM user that allows the user to pass only those approved roles. You usually add `iam:GetRole` to `iam:PassRole` so the user can get the details of the role to be passed. In this example, the user can pass only roles that exist in the specified account with names beginning with `EC2-roles-for-XYZ-` :

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
              "iam:GetRole",
              "iam:PassRole"
          ],
          "Resource": "arn:aws:iam:: 111122223333 :role/EC2-roles-for-XYZ-*"
      }
  ]
}
```

Now the user can start an Amazon EC2 instance with an assigned role. Applications running on the instance can access temporary credentials for the role through the instance profile metadata. The permissions policies attached to the role determine what the instance can do.

Example 2

Amazon Relational Database Service (Amazon RDS) supports a feature called **Enhanced Monitoring**. This feature enables Amazon RDS to monitor a database instance using an agent. It also allows Amazon RDS to log metrics to Amazon CloudWatch Logs. To enable this feature, you must create a service role to give Amazon RDS permissions to monitor and write metrics to your logs.

To create a role for Amazon RDS enhanced monitoring

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. Choose **Roles**, and then choose **Create role**.
3. Choose the **AWS Service** role type, and then for **Use cases for other AWS services**, choose the **RDS** service. Choose **RDS – Enhanced Monitoring**, and then choose **Next**.
4. Choose the **AmazonRDSEnhancedMonitoringRole** permissions policy.
5. Choose **Next**.
6. For **Role name**, enter a role name that helps you identify the purpose of this role. Role names must be unique within your AWS account. When a role name is used in a policy or as part of an ARN, the role name

is case sensitive. When a role name appears to customers in the console, such as during the sign-in process, the role name is case insensitive. Because various entities might reference the role, you can't edit the name of the role after it is created.

7. (Optional) For **Description**, enter a description for the new role.
8. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tags for AWS Identity and Access Management resources.
9. Review the role and then choose **Create role**.

The role automatically gets a trust policy that grants the `monitoring.rds.amazonaws.com` service permissions to assume the role. After it does, Amazon RDS can perform all of the actions that the `AmazonRDSEnhancedMonitoringRole` policy allows.

The user that you want to access Enhanced Monitoring needs a policy that includes a statement that allows the user to to list the RDS roles and a statement that allows the user to pass the role, like the following. Use your account number and replace the role name with the name you provided in step 6.

```
 {
  "Sid": "PolicyStatementToAllowUserToListRoles",
  "Effect": "Allow",
  "Action": ["iam:ListRoles"],
  "Resource": "*"
},
{
    "Sid": "PolicyStatementToAllowUserToPassOneSpecificRole",
    "Effect": "Allow",
    "Action": [ "iam:PassRole" ],
    "Resource": "arn:aws:iam:: account-id :role/ RDS-Monitoring-Role "
}
```

You can combine this statement with statements in another policy or put it in its own policy. To instead specify that the user can pass any role that begins with `RDS-`, you can replace the role name in the resource ARN with a wildcard, as follows.

```
    "Resource": "arn:aws:iam:: account-id :role/RDS-*"
```

`iam:PassRole` actions in AWS CloudTrail logs

`PassRole` is not an API call. `PassRole` is a permission, meaning no CloudTrail logs are generated for IAM `PassRole`. To review what roles are passed to which AWS services in CloudTrail, you must review the CloudTrail log that created or modified the AWS resource receiving the role. For example, a role is passed to an

AWS Lambda function when it's created. The log for the `CreateFunction` action shows a record of role that was passed to the function.
