Overview

A JSON policy document in which you define the principals that you ***trust*** to assume the role.

A role trust policy is a required resource-based policy that is attached to a role in IAM.

The principals that you can specify in the trust policy include **users, roles, accounts, and services.**

An example of a simple trust policy

A common use case is when you need to provide security audit access to your account, allowing a third party to review the configuration of that account.

After attaching the relevant permission policies to an IAM role, you need to add a cross-account trust policy to allow the third-party auditor to make the sts:AssumeRole API call to elevate their access in the audited account.

The following trust policy shows an example policy created through the AWS Management Console:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

Note: The suffix *root* in the policy's Principal attribute equates to "authenticated and authorized principals in the account," not the special and all-powerful root user principal that is created when an AWS account is created.

As you can see, it has the same structure as other IAM policies with Effect, Action, and Condition components.

It also has the **Principal parameter, but no Resource attribute**.

This is because the resource, in the context of the **trust policy, is the IAM role itself.**

For the same reason, the Action parameter will only ever be set to one of the following values: **sts:AssumeRole, sts:AssumeRoleWithSAML, or sts:AssumeRoleWithWebIdentity.**

Using the Principal attribute to reduce scope

In a trust policy, the Principal attribute indicates which other principals can assume the IAM role. In the example above, 111122223333 represents the AWS account number for the auditor's AWS account. In effect, this allows any principal in the 111122223333 AWS account with sts:AssumeRole permissions to assume this role.

To restrict access to a specific IAM user account, you can define the trust policy like the following example, which would allow only the IAM user LiJuan in the 111122223333 account to assume this role. LiJuan would also need to have sts:AssumeRole permissions attached to their IAM user for this to work:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:user/LiJuan"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

Trust policies for AWS services that assume roles

There are two types of contexts where AWS services need access to IAM roles to function:

1. Resources managed by an AWS service (like Amazon EC2 or Lambda, for example) need access to an IAM role to execute functions on other AWS resources, and need permissions to do so.
2. An AWS service that abstracts its functionality from other AWS services, like Amazon Elastic Container Service (Amazon ECS) or Amazon Lex, needs access to execute functions on AWS resources. These are called service-linked roles and are a special case that's out of the scope of this post.

Here's an example trust policy for a role designed for an Amazon EC2 instance to assume. You can see that the principal provided is the ec2.amazonaws.com service:

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

NOTE:

1. Several customers have asked if it's possible to design a trust policy for an IAM role such that it can only be passed to a specific Amazon EC2 instance. This isn't directly possible. **You cannot place the Amazon Resource Name (ARN) for an EC2 instance into the Principal of a trust policy, nor can you use tag-based condition statements in the trust policy to limit the ability for the role to be used by a specific resource.**

The only option is to manage access to the iam:PassRole action within the permission policy for those IAM principals you expect to be attaching IAM roles to AWS resources. This special Action is evaluated when a principal tries to attach another IAM role to an AWS service or AWS resource.

You should use restrictions on access to the iam:PassRole action with permission policies and permission boundaries. This means that the ability to attach roles to instance profiles for Amazon EC2 is limited, rather than using the trust policy on the role assumed by the EC2 instance to achieve this. This approach makes it much easier to manage scaling for both those principals attaching roles to EC2 instances, and the instances themselves.

You could use a permission policy to limit the ability for the associated role to attach other roles to Amazon EC2 instances with the following permission policy, unless the role name is prefixed with EC2-Webserver-:

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

https://aws.amazon.com/blogs/security/how-to-use-trust-policies-with-iam-roles/