AWS IAM Policies

Contents

Overview

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) 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 IAM principal (user or role) makes a request.
Permissions in the policies determine whether the request is allowed or denied.

For example, if you attach the policy that provides access to AWS EC2 to the user John, and you do not attach the same policy to the user David, then whenever John sends a request to AWS EC2 it will be accepted. However, when David sends request to AWS EC2 it will be rejected, as David does not have the correct policy attached to make request to AWS EC2.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API. This means that, regardless of the type of access a user or role has (e.g. programmatic, web console, etc.), they can send requests to any services within AWS depending on the policies attached to that user or role.

AWS Support 6 types of policies:

- Identity-based
- Resource-based
- Permissions boundaries
- Access Control Lists (ACLs)
- Session Policies

Identity-based

The identity-based policy is the one that can be attached directly to AWS identities like user, group or roles.
These policies can be AWS managed or a customer-managed. The main difference is that AWS managed are policies which AWS created, and customer-managed are custom policies created by users in the AWS environment.

Resource-based

Resource-based policies are JSON policy documents that you attach to a resource such as an Amazon S3 bucket.
With these policies, you can specify who has access to a resource and what action they can perform on that resource.

Permissions boundaries

A permissions boundary is an advanced feature in which you set the maximum permission that a user or role can have access to.
When you set a permissions boundary for an entity, the entity can perform only the actions that are allowed by both its identity-based policies and its permissions boundaries.
Explicit denies can be set in the permissions boundary to deny actions that may be allowed by the policy.
For example, if you had a user that was given Administrative Access, and a permissions boundary was set to deny the creation of IAM users, then the user will be unable to create a user.

Session Policies

Session policies are advanced policies that you pass in a parameter when you programmatically create a temporary session for a role or federated user.
An example is that if a user has S3 and EC2 full access, and during an assume role action which creates a temporary session, we allow this session to have S3 full access. For this session, the user will have full S3 access but will be denied access to EC2 resources, as we limited the permission for this session.

IAM Policy Structure

There are two ways you can create IAM policies from IAM web console. Visual Editor and a character-based JSON policy editor.
This section will cover how to create policies in JSON format.
Creating custom IAM policy in JSON format, gives more fine-grained control over what permissions can be granted.

There are 4 elements when creating an IAM Policy:

- Effect
- Action
- Resource
- Condition

Effect

You determine whether the policy **Allow** or **Deny**.
These are the only options.

Action

This contains a list of actions, for example, to Read or to Write.
This corresponds with the *Effect* element, as in, you allow the specific action described.

Resource

If you create an IAM permissions policy, you must specify a list of resources to which the actions apply. If you create a resource-based policy, this element is optional. If you do not include this element, then the resource to which the action applies is the resource to which the policy is attached.

Condition

Specify the circumstances under which the policy grants permission.

Example

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "logs:CreateLogGroup",
            "Resource": "arn:aws:logs:eu-west-2:735130171699:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:eu-west-2:735130171699:log-
group:/aws/lambda/post_lambda:*"
            ]
        }
    ]
}
```

IAM Policy Evaluation

Evaluating IAM policies, by default, all requests are denied.
A user with no policies attached that allows access to any resources will, by default have a denied access to all resources.
An explicit *Allow*, overrides the default settings that denies.
And an explicit *Deny*, overrides the *Allow*

Tutorial

No Tutorial for this module.

Exercises

No Exercises for this module.