COURSEWARE

AWS IAM Role CLI

Contents

Overview

Identity Access Management (IAM) is a service which can allow the root user to create accounts, roles and groups.
The root user can also grant permissions and to a certain degree of element.
So, if a developer needs access to an EC2 instance, rather than providing the credentials for
the root user, the root user will create an account which has only access to that EC2 instance.
This is known as **Principle of least privilege**.

In this section you will be guided on how to create a role for EC2 instances to have access
to RDS through the AWS CLI.

You will need to have a machine configured with AWS CLI, preferably with administrative permissions.

Tutorial

1. We need to know what available AWS managed policies there are:

```
aws iam list-policies --scope AWS
```

If you know the name of the policy, then you can filter the list by running:

```
aws iam list-policies --scope AWS | grep (PolcyName Insert here)
```

This command will output the **PolicyName** and **arn**. In the case of finding the Admin
policy, the output is as follows:

Example command:

```
aws iam list-policies --scope AWS | grep AdministratorAccess
```

```
"PolicyName": "AdministratorAccess",
"Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
```

2. When using the AWS CLI to create a role you will need to create and attach roles and
   permissions separately.

3. When creating a role through AWS CLI, you will need to create a policy that has the
   "assumeRole" action. We need this because if this role requires additional roles to be associated with it, then we have the ability to do so.

4. Create a policy called **"assume-policy.json"**. Run the following commands:

```
ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
cat << EOF > assume-policy.json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": { "AWS": "arn:aws:iam::${ACCOUNT_ID}:root" },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
```

The *ACCOUNT_ID*, will have the value of the main account holders id. This is needed as we
require this unique id as part of the principal.

5. Now we can create a role. Remember you must be in the same directory as your policy. Run
   the following command:

```
aws iam create-role --role-name test-role --assume-role-policy-document file://
assume-policy.json
```

6. If you navigate to your AWS account -> IAM -> Roles, you should see a new role called

test-role. This role will have no permissions associated with it, so lets add one.

We are going to add a simple **EC2FullAccess** permission, run the following command:

```
aws iam attach-role-policy --role-name test-role --policy-arn
arn:aws:iam::aws:policy/AmazonEC2FullAccess
```

We have listed all the AWS managed policies in step 1, alternatively you can go to your AWS
Account -> IAM -> Policies, and look up the policies and copy its ARN.

If you navigate to AWS -> IAM -> Roles, and look at the permission for *test-role* it will have *EC2FullAccess* policy attached.

7. The role creation part is now done, we need to know how we can clean up after.
   Firstly we detach the policies from the IAM role created, as AWS CLI will not
   allow you to delete a role with policies attached.

Run this command to detach the policy we attached to *test-role*:

```
aws iam detach-role-policy --role-name test-role --policy-arn
arn:aws:iam::aws:policy/AmazonEC2FullAccess
```

8. We can now delete the role we created by running:

```
aws iam delete-role --role-name test-role
```

Exercises

1. Create a role through AWS CLI, call it **EC2RDSRole**, and give this role

RDSFullAccess