Running AWS Services with Restricted User Accounts

In this lesson, you will explore how to run AWS services with restricted user accounts.

WE'LL COVER THE FOLLOWING

For trying out examples in this course, it's easiest to use access keys assigned to a user with full access to all AWS resources. If you created access keys as suggested in the previous lesson, those keys will have full access to all your resources, so you can skip this lesson.

Restricted privileges account

If you would like to set up an account with restricted privileges, the user account will need the following policies:

- `arn:aws:iam::aws:policy/AWSLambdaFullAccess`
- `arn:aws:iam::aws:policy/AmazonAPIGatewayAdministrator`
- `arn:aws:iam::aws:policy/IAMFullAccess`

Access to CloudFormation

In addition, the user account will need access to CloudFormation. There are no standard AWS policies for this, so you will need to create a new policy and assign it to the user account. The JSON template below provides full access to all CloudFormation resources:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": "cloudformation:*",
      "Resource": "*"
    }
  ]
}
```

Some companies restrict user accounts for developers, in which case you may need to ask your administrator to assign the policies mentioned.

If full access to IAM and CloudFormation is a problem, ask the administrator to create a separate sub-account for you in your AWS account organization (the administrator can do so from the AWS Organisations console). Each sub-account has completely isolated resources, so granting full access to a subaccount does not assign any privileges to important corporate resources.

Using a profile

AWS command-line tools can store multiple combinations of access keys on the same system. This is a convenient way to use separate access credentials for different projects or to reduce the chance of mistakes by restricting everyday usage to read-only access. A key combination is called a *profile*.

If your IT administrator creates a subaccount for experiments that is different from your main AWS account, you will most likely want to record the keys in a separate profile in order to easily switch between access combinations. If your account is managed by a company and the IT security department does not want to give you the required access to try out SAM, you can register a personal account with AWS and set it up as a separate profile.

Setting up a profile

You can set up a profile by adding the `--profile` option to the command for configuring access keys, followed by a profile name. For example, the following command will help you create a profile called `samdevelop`:

```
aws configure --profile samdevelop
```

To use a specific profile, add the `--profile` argument to all the AWS and SAM commands. For example, execute the following command to test whether the profile `samdevelop` is set up correctly:

```
aws sts get-caller-identity --profile samdevelop
```

To keep things simple, the process of profile setting will be omitted from the rest of the course. If you would like to use a separate profile, remember to add it to all the command line examples starting with `aws` or `sam`.

Now that the tools are ready, you'll put them to good use by creating a Lambda function.

The next lesson lists some interesting experiments. You can perform these to get a better understanding of what you have seen so far.