Cloud Computing Exercise #17

Creating an EC2 role for AWS ser vice access

A. Preparation

1. Sign in to your AWS account as the non-root admin user.
2. In this lab, you will create a role for an EC2 instance, start an EC2 instance, associate this instance with the role, and then run a script on the EC2 instance that will use AWS services to stop the instance automatically after 2 minutes.

B. Create a new IAM role

1. Go to IAM Roles (IAM/Access management/Roles) and select “Create role”. The trusted entity type should be “AWS Service” (the default), and the “Use case” should be set to “EC2”. Go to the next page.
2. Open a new tab and navigate to Policies and Select “Create policy “and click on the “JSON” tab to enter the policy for the role. There should be a single statement that will allow the EC2 instance associated with this role to stop itself:
   * The action string should be: “ec2:StopInstances”,
   * The resource should be: all resources (user wildcard),
   * We do not want this instance to be able to stop any instances, so we need to add a condition to this policy that allows the instance to stop only those instances that have been tagged with the resource tag key “stoppable” and value “true”. Add the following IAM condition to the policy statement:

"Condition": { "StringEquals": { "ec2:ResourceTag/stoppable": "true" } }

1. Give the new policy the name “ec2\_allow\_stop\_policy” and create the policy. The new policy should appear in the list of IAM policies (IAM/Access management/policies).

(If you need assistance with structuring the policy, you can check the modules section for guidance.)

1. Go back to the role creation tab (“Create role” page) and find the policy you just created. You can use the search bar to find the policy “ec2\_allow\_stop\_policy”. Select this policy by checking its checkbox, and attach it to the new role (click “Next”). Give the new role the name “ec2\_allow\_stop\_role”. The new role should show up in the list of roles in your account (IAM/Access management/Roles).

C. Launch an EC2 instance

1. Go to EC2 and launch a new instance using the “Amazon Linux 2023 AMI” with the t2.micro instance type.

* In the “Name and Tags” step, add a tag with the key “stoppable” and value “true.” This tag ensures compliance with the IAM policy condition. Make sure this tag is applied only to the EC2 instance and not to the EBS volume or network interface by unchecking the corresponding boxes.
* In the “Advanced details” section, set the IAM instance profile to the role “ec2\_allow\_stop\_role.” Then, scroll down to the metadata version settings and set it to V1 and V2 (token optional). This is important because our upcoming script will need the instance ID to shut down the instance. Since AWS requires security tokens for IMDSv2, enabling V1 will simplify this exercise and allow our script to work correctly. Lastly, set the metadata response hop limit to 64 and select “Enabled” for allow tags in metadata, and “Enabled” for Metadata accessible.
* The default security group settings are sufficient, allowing SSH access from anywhere. Finally, attach your key pair, launch the EC2 instance, and connect to it using your local SSH client.

D. Install the necessary software on the EC2 instance

1. Install the unzip utility on your EC2 instance. Type: sudo yum install unzip
2. Download the AWS command line interface (CLI) utility using curl.

Type: curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"

1. Unzip the archive file in your home directory. Type: unzip awscliv2.zip
2. Install the AWS CLI. Type: sudo ./aws/install

You should have the AWS CLI installed on your EC2 instance. You can check if the installation was completed correctly by typing: aws --version

This should show you the AWS CLI version and the associated Python interpreter and Linux versions.

1. Install the nano text editor. Type: sudo yum install nano
2. Install the “at” task scheduling utility. Type: sudo yum install at
3. Start “atd”, the task scheduling daemon process. Type: sudo systemctl start atd

E. Create and run a Bash shell script that terminates the EC2 instance after 2 minutes

1. Start nano and enter the following shell script:

#! /bin/bash -ex

INSTANCEID=$(curl -s http://169.254.169.254/latest/meta-data/instance-id)

echo "aws --region us-east-1 ec2 stop-instances --instance-ids $INSTANCEID" \

| at now + 2 minutes

Copy the script as it is. You only need to update the region if your instance is launched outside of us-east-1. Change it to the region where your instance is running.

Shell scripts are beyond the scope of this course, but this is basically what is going on:

* The first line specifies that the script should be run by the bash interpreter.
* The second line accesses the AWS EC2 instance meta data services using the curl command, gets the EC2 instance’s own instance ID and saves it in the INSTANCEID variable.
* The third line uses the AWS CLI to issue a “stop instance” command to the instance specified by the INSTANCEID variable (thus, the instance stops itself). However, the command is not issued immediately: the at utility schedules the command 2 minutes in the future. Note that the third and the fourth lines form one “logical” line; this is why the “\” character is necessary at the end of the third line.

1. Save the file as myscript.sh, and change its file permissions such that the owning user (you) could read, write and execute this file (chmod +x myscript.sh). Then, run the script by entering its file name.

Type: ./myscript.sh

1. You should see the commands of the script printed on the screen as they are executed. Check your instance in the management console – it should stop after two minutes. Also, you should get an error message that you have lost the SSH connection with your instance. You can go to the management console, start your instance again, connect to it, and run the shell script again if you wish. The same thing will repeat.

F. Clean up after yourself

1. Go to the list of EC2 instances, and terminate your instance. Go to IAM roles (IAM/Access management/Roles) and delete the role named ”ec2\_allow\_stop\_role”. Go to IAM policies (IAM/Access management/Policies) and delete the policy named “ec2\_allow\_stop\_policy”.
2. Log out of AWS.