*Create and Manage Users with AWS IAM

06:07 AM

Create Groups Based on Required Access

In the top search bar, type in and click on IAM.

In the left-hand menu, click User groups.

Click Create group to enter the Create user group wizard.

Set the User group name to Developer.

In the Attach permissions policies section, type AmazonEC2FullAccess into the search box and then press enter. Check the box next to the AmazonEC2FullAccess policy to attach it.

You will now attach a second policy, so first, click Clear filters. Then type AWSCodeBuildDeveloperAccess, and check the resulting policy.

Click Create group.

Repeat the process above to create these four additional groups:

Group Name: Administrators

Policies: AdministratorAccess

Group Name: DevOpsEngineer

Policies: AmazonEC2FullAccess, AmazonS3FullAccess

Group Name: Security

Policies: AWSSecurityHubFullAccess, SecurityAudit

Group Name: ITManagement

Policies: Billing

Once you're done, you'll see all five groups listed at the User groups page.

With all of the groups created, you now are ready to create users!

Create Users and Apply Groups

With the global account settings configured and groups created for the different access requirements, it is time to populate IAM with users.

Generally, there will be a required set of information from each user to ensure you can create the users with the proper tags and permissions. The information provided for users that need access to resources in this environment is in this table:
figure
Here are the emails. Note that they include the names and teams from the table above from which you can copy and paste for the tasks below:

Martyn.Wilkins@globomantics.com

AWS Administrators

Asmaa.Squires@globomantics.com

Developers

Allegra.Mcgill@globomantics.com

Dev Ops Engineers

Warwick.Ahmad@globomantics.com

IT Management

Note: The row for your name, on the Security team, was excluded.

You will now create those five users, one for each group.

From the left-hand menu click Users, then click Add users.

Note: This is at the Identity and Access Management (IAM) page where you ended the previous Challenge.

Starting with Martyn, enter the following using the table above.

User name: Martyn.Wilkins (This is the first part of the email)

Access key - Programmatic access: since the table says Yes, check this

box.

Password - AWS Management Console access: All users in this lab will have this box checked.

Leave Autogenerated password and User must create a new password at next sign-in checked.

Click Next: Permissions

Select the checkbox for the Administrators group, and click Next: Tags.

In the Key section, type Email.

For the Value paste in the provided email for Martyn, from above.

Type in the next Key called Team, and enter a Value of AWS Administrators.

Click Next: Review and review the user details to ensure they are correct, then click Create user.

Note: In the Success banner that appears on the Add user page, you can see the sign-in link to the AWS Management Console, and you can download the access keys. This will be the only time that you can download these access keys, and best practice for the Security Access manager is to keep a record of these keys offline. This is not required for this lab.

You will not do this in this lab, but normally you would now send the user their new account information by clicking Send email at the far right of the table.

This will work only if AWS simple email service is properly set up. SES is not covered in this lab. This would email the user their access information as well as the link required to login to the organization’s console. Here is a sample email of what would be sent:

Hello,
You have been given access to the AWS Management Console for the Amazon Web Services account with the ID ending in 2825. You can get started by using the sign-in information provided below.

Sign-in URL: https://676287162825.signin.aws.amazon.com/console
User name:

Your initial sign-in password will be provided to you by your AWS account

administrator, separately from this email. When you sign in for the first time, you must change your password.

Stay connected with AWS by creating a profile:
https://pages.awscloud.com/IAM-communication-preferences.html

Sincerely,

Your AWS Account Administrator

You would need to also separately send the user their temporary password.

Click Close.

Repeat the above steps for each of the other four users (one for each Group). Pay attention to assigning the correct Group for each, based on their team, and if they should be given programmatic access.

Remember you will be one user, and you will be in the Security Group, like Aaron in the screenshot below.

Once complete you will have five users including yourself.
figure
Note:

You will see a sixth user which corresponds to the pluralsight- prefixed user provided by this lab.

In the real world, every company is going to have required information for each user. Tags are where you enter that information. It is important that you maintain consistency with a policy for each user created, which is another good reason to lock administrator privileges down to only a select group that requires the ability to create accounts.

When creating each user you could, on the Add user page, click Download .csv to add these credentials to an offline encrypted drive used in case they are required for a business-critical function later. Some examples are to recover work done after an employee is terminated, or for legal forensic investigations should this employee be suspected of illicit activity.

Create A Custom Policy Access To S3 Bucket Functionality

n this challenge, there are two resources that comprise the initial part of the application that is being migrated to AWS: an EC2 Instance, and an S3 Bucket. Both of these have already been created.
The EC2 instance is running an application that needs to be able to read the data from the S3 bucket without a user logged in. Further, you do not want the EC2 instance to be able to access any other resources or S3 buckets beside the one specified, and you most certainly do not want any other EC2 instances to be able to access the S3 bucket.

Your company is highly concerned about access to the sensitive data in this S3 bucket. S3 bucket data compromise has been in the news lately, and are are a prime concern for management.

This EC2 instance needs to be assigned a role, and that role needs to have specific policies attached that enable appropriate granular control. So, much like creating groups to then assign to users, you need to create a policy you will assign to a role.

From the left-hand menu click Policies.

Click the Create Policy button.

Click Choose a service, type in then and click on S3.

Under Access level, select Read.

Expand the Resources section, leave Specific selected, and under bucket, click Add ARN.

In the Add ARN(s) pop-up, for the Bucket name enter sensitive-globo-bucket-* (this will append to the ARN prefix, so the entire ARN will read arn:aws:s3:::sensitive-globo-bucket-*), then click Add.

Note: This bucket was created for you when you started this lab.

For accesspoint, job, and object, check the Any box.
figure

Note: The above screenshot will probably be a bit different, but do note there will be resources for which you are not checking anything, such as the multiregion ones.

Expand the Request conditions section and check Source IP, then enter the private IP of the EC2 instance 172.31.37.38 (This EC2 was created for you

when you started the lab.)

Click Next: Tags, then click Next:Review.

At the Create policy page, enter the policy Name of S3-Sensitive-Appdata, and a Description of Restricting access to S3 bucket with customer data by ip.

Then click Create policy.

You will see a S3-Sensitive-Appdata has been created message at the top of the page.

All done! The policy is now created and ready to be applied to a role.

Create a Custom Role for EC2 Instance Access to S3 Bucket

A policy is no use without applying the policy to a role, and that role to the requesting resource. Roles are used to assign temporary permissions to users, applications, or services that don't have permanent credentialed access to your AWS resources. For example, these entities can be users from other accounts.
Or in this case an EC2 instance that is running an application that needs access to read the sensitive data on the S3 bucket.

From the left-hand menu click Roles.

Click Create role to begin.

Leave the type of trusted entity as AWS service. Under Choose a use case, select EC2.
figure
In the lower-right click Next: Permissions.

Type S3 in the search box, and check the box next to the S3-Sensitive-Appdata policy (you may need to scroll down a bit to see it). Then click Next: Tags to continue.

Note: This will attach the previously created policy.

Create a tag with a Key called type, and a Value of service. Then click Next: Review.

Enter a Role name of EC2-read-sensitiveS3, leave the Role description untouched, and click Create role.

Note: Now that the role is created, you need to apply it to the resource that needs to assume the role. In this case it is the EC2 instance, and you'll do that shortly.

At the top of the page in the search bar, type in and click on EC2.

From the left-hand menu under the Instances section, click Instances.
figure

You may not see any instances. If not, it is because AWS has changed your region. Toward the upper-right, click the drop-down to the left of the pluralsight- user name you logged in with, and choose US West (Oregon) us-west-2.
figure

On the Instances page, select the checkbox for the sole instance listed.

Click Actions > Security > Modify IAM role.
figure

Click the IAM role drop-down, and select EC2-read-sensitiveS3.

Click Save.

Back at the EC2 Instances page, select the instance again (if needed), and at the bottom of the page, in the Security tab, you will see the IAM Role value reflecting the applied EC2-read-sensitiveS3 policy. If you do not, click the refresh button, select the EC2 instance, and go to the Security tab again.

This completes the assignment of the role and associated permissions at the granular level required for the safety of sensitive data. Should the application need to scale, you can change the specific IP to a range of IPs within a protected Security Group, and apply this same role to multiple EC2 instances as required.