← Back to the **Course**

Level: Advanced

AWS Certified Security Specialty

Identity and Access Management -Quiz

Completed on **Sun, 27 Feb 2022**

1st	0/5	0.00%
Attempt	Marks Obtained	Your Score

0h 0m 6s	FAIL
Time Taken	Result

Domain wise Quiz Performance Report

Join us on Slack community

No.	Domain	Total Question	Correct	Incorrect	Unattempted
1	Identity and Access Management	1	0	0	0
2	Other	4	0	0	0
Total	**All Domains**	**5**	**0**	**0**	**0**

Review the Answers

Filter By All Questions

Question 1

Unattempted

Domain: Other

A company has multiple AWS Accounts. They temporarily need to ensure users from a production-based account can access a staging account. Which of the following is the right way to ensure this access is put in place?

A. Create an IAM user with Access keys and the right access policies in the staging account.

B. Copy the IAM user from the production-based account to the staging account.

C. Create a Cross-Account IAM Role in the staging account that can be assumed by the production account. right

D. Create a Cross-Account IAM Role in the production account that can be assumed by the staging account.

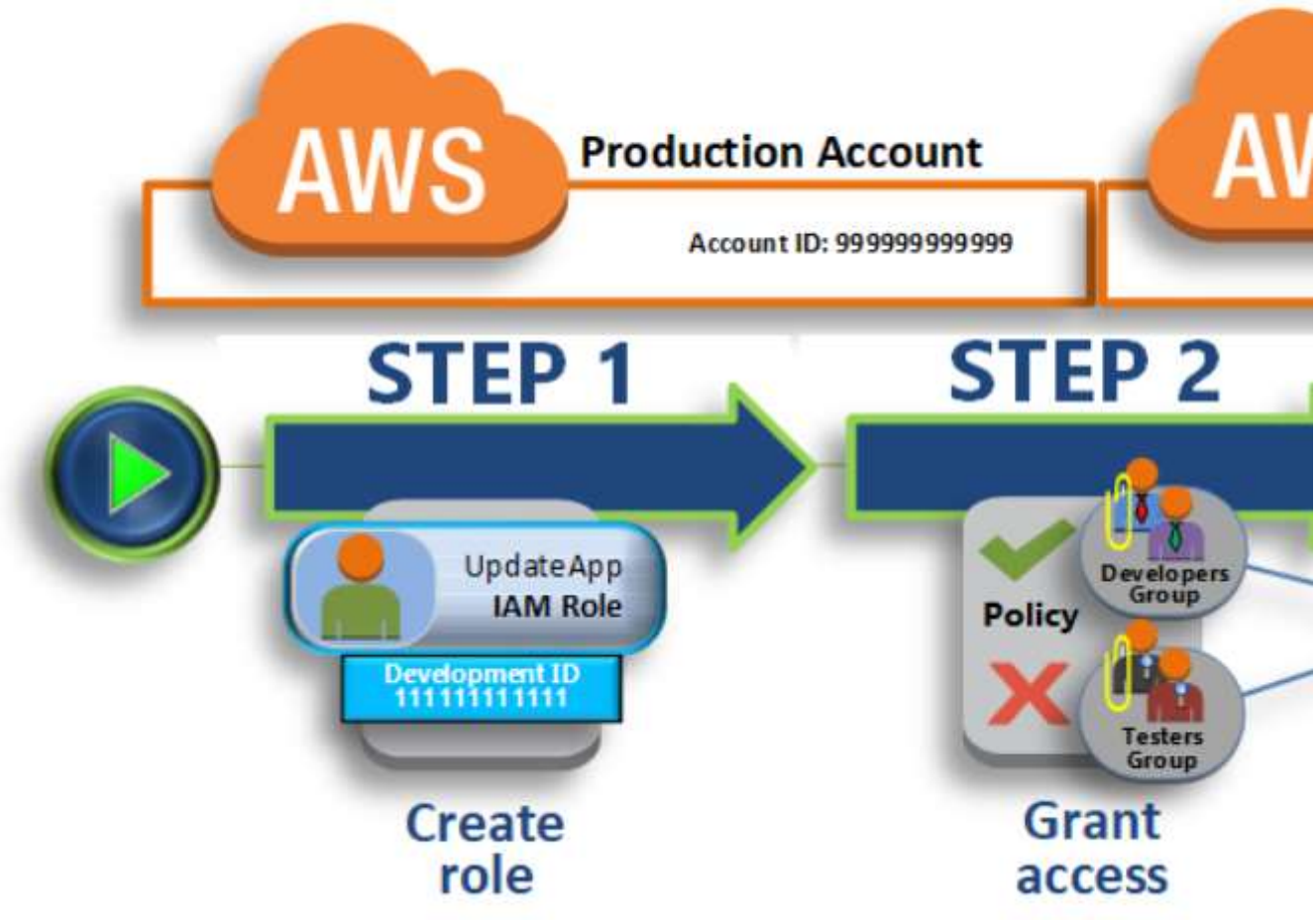
Explanation:

Answer - C

This workflow example is also given in the AWS Documentation.

You share resources in one account with users in a different account. By setting up cross-account access in this way, you don't need to create individual IAM users in each account. In addition, users don't have to sign out of one account and sign in to another to access resources in different AWS accounts. After configuring the role, you see how to use the role from the AWS Management Console, the AWS CLI, and the API.

This workflow has three basic steps.



Because this is clearly mentioned in the AWS Documentation, all other options are invalid.

For more information on cross-account access with roles, please visit the below URL

https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html

Ask our Experts

View Queries

Question 2 Unattempted

Domain: Other

Your company has just started using an AWS account. They want to ensure that they apply the right security principles to the root user of their AWS Account. Which of the following are the right security measures to put in place? Choose 2 answers from the options given below.

A. Ensure that the root account is used for privileged daily account activities.

B. Delete the AWS root account access keys if you no longer use them. right

C. Have a rotation policy in place for changing the IAM passwords for IAM users. right

D. Create Access keys and provide them to the respective IT Administrators.

Explanation:

Answer – B and C

This is mentioned in the AWS Documentation.

We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

You can create, rotate, disable, or delete access keys (access key IDs and secret access keys) for your AWS account root user. You can also change your root user password. Anyone with root user credentials for your AWS account has unrestricted access to all the resources in your account, including billing information.

Options A and D are incorrect since these are not the best security practices for the root user account.

For more information on the root user, please visit the below URL

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html

Ask our Experts

View Queries

Question 3

Unattempted

Domain: Other

A company has a string of AWS Accounts and several IAM users defined in each account. For auditing purposes, they need to ensure that all calls to AWS IAM are logged. How can they achieve this?

A. Use the AWS Config service.

B. Use the AWS Inspector service.

C. Use the AWS Cloudwatch service.

D. Use the AWS Cloudtrail service. right

Explanation:

Answer – D

This is mentioned in the AWS Documentation.

For an ongoing record of events in your AWS account, including events for IAM and AWS STS, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all regions. The trail logs events from all regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify.

Option A is invalid since this service is used to check for configuration changes.

Option B is invalid since this service is used to check for vulnerabilities in AWS resources.

Option C is invalid since this service is used for logging purposes.

For more information on Cloudtrail integration, please visit the below URL

https://docs.aws.amazon.com/IAM/latest/UserGuide/cloudtrail-integration.html

Ask our Experts

View Queries

Question 4

Unattempted

Domain: Other0

A development team has developed a Java application that makes requests to a DynamoDB table. The application needs to be deployed on an Auto Scaling group launched through a launch template. Which of the following is the right way to ensure the application can access the DynamoDB table properly?

A. Add the IAM Access Keys in the environment variables of the EC2 Instance.

B. Embed the IAM Access Keys in the application.

C. Create an IAM Role with the right permissions and configure the role in the launch template of the ASG. right

D. Create an IAM Role with the right permissions and attach the role to the instances launched in the Auto Scaling group.

Explanation:

Answer - C

This is mentioned in the AWS Documentation.

Applications must sign their API requests with AWS credentials. Therefore, if you are an application developer, you need a strategy for managing credentials for your applications that run on EC2 instances. For example, you can securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests while protecting your credentials from other users. However, it's challenging to securely distribute credentials to each instance, especially those that AWS creates on your behalfs, such as Spot Instances or instances in Auto Scaling groups. You must also be able to update the credentials on each instance when you rotate your AWS credentials.

We should use IAM roles so that your applications can securely make API requests from your instances without requiring you to manage the security credentials.

Option C is CORRECT because, in the launch template, you can configure the IAM role used by the ASG instances.

For more information on IAM Roles for EC2, please visit the below URL

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html

Ask our Experts

View Queries

Question 5

Unattempted

Domain: Identity and Access Management

Your IT Security Administrator has defined the following policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "152.0.2.0/24"
          ]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

What does the following policy define?

A. Allows access to all AWS resources from workstations in the IP range of 152.0.2.0/24. right

B. Allows access to all AWS resources from the workstation with an IP of 152.0.2.0.

C. Allows access to all AWS resources except from workstations in the IP range of 152.0.2.0/24.

D. Allows access to all AWS resources except from workstation with an IP of 152.0.2.0.

Explanation:

Answer - A

This example shows how you might create a policy that denies access to all AWS actions in the account when the request comes from outside the specified IP range.

AWS: Denies Access to AWS Based on the Source I

This example shows how you might create a policy that denies access to all A from outside the specified IP range. The policy is useful when the IP addresse This policy also provides the permissions necessary to complete this action or in the example policy with your own information.

The `aws:SourceIp` condition key denies access to an AWS service, such as AW For more information about using the `aws:SourceIp` condition key, see AWS

Important

This policy does not allow any actions. Use this policy in combination w

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NotIpAddress": {
                "aws:SourceIp": [
                    "192.0.2.0/24",
                    "203.0.113.0/24"
                ]
            }
        }
    }
}
```

By default, all requests are denied, so you must provide access to the services, actions, and resources that you intend for the identity to access. If you also want to allow access to complete the specified actions in the IAM console, you need to provide additional permissions.

With this policy, the allow statement allows everything, and the deny statement is used with the condition to deny every access that is matching the condition. In the condition block, the `NotIpAddress` condition is provided a key-value pair for evaluation. In this policy, it is using the `aws:SourceIp` AWS-wide key. So if the source IP is not in the IP range of **152.0.2.0/24 then the request is denied as per the policy.**

But if it is within that range, it will be allowed as per the allow statement within the policy.

For more information on example IAM policies, please visit the below URL

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_examples.html

Ask our Experts

View Queries

Finish Review