1. A Solutions Architect is designing a shared service for hosting containers from several customers on Amazon ECS. These containers will use several AWS services. A container from one customer should not be able access data from another customer. Which of the below solutions should the architect use to meet these requirements?

   a: IAM Role for Tasks
   b: IAM Role for EC2 instances
   c: IAM instance profile for EC2 instances
   d: Security Group Roles

2. You are deploying an application on Amazon EC2, which must call AWS APIs. What method should you use to securely pass credentials to the application?

   A. Pass API credentials to the instance using Instance user data.
   B. Store API credentials as an object in Amazon S3.
   C. Embed the API credentials into your application.
   D. Assign IAM roles to the EC2 Instances.

3. Which of the following is not a feature of AWS Security Token Service?
   A. STS enables you to request temporary, limited-privilege credentials.
   B. STS enables users to assume role.
   C. STS generate Git Credentials for IAM users.
   D. STS generates Federated Credentials for IAM users.

4. Your organization AWS Setup has an AWS S3 bucket which stores confidential documents which can be only downloaded by users authenticated and authorized via your application. You do not want to create IAM users for each of these users and as a best practice you have decided to generate AWS STS Federated User temporary credentials each time when a download request is made and then use the credentials to generate pre-signed URL and redirect user for download. However, when user is trying to access the pre-signed URL, they are getting Access Denied Error. What could be the reason?

   A. AWS STS service must be given access in S3 bucket ACL.
   B. IAM User used to generate Federated User credentials does not have access on S3 bucket.
   C. IAM Role used to generate Federated User credentials does not have access on S3 bucket.
   D. Your application must be whitelisted in AWS STS service to perform Federated User action.

5. Your organization has an AWS setup and planning to build Single Sign On for users to authenticate with on-premise Microsoft Active Directory Federation Services (ADFS) and let user's login to AWS console using AWS STS Enterprise Identity Federation. Which of the following service you need to call from AWS STS service after you authenticate with your on-premise?

   A. AssumeRoleWithSAML
   B. GetFederationToken

C. AssumeRoleWithWebIdentity
D. GetCallerIdentity

6. An EC2 Instance hosts a Java based application that accesses a DynamoDB table. This EC2 Instance is currently serving production users. Which of the following is a secure way for the EC2 Instance to access the DynamoDB table?

A. Use IAM Roles with permissions to interact with DynamoDB and assign it to the EC2Instance.
B. Use KMS Keys with the right permissions to interact with DynamoDB and assign it tothe EC2 Instance.
C. Use IAM Access Keys with the right permissions to interact with DynamoDB and assign it to the EC2 Instance.
D. Use IAM Access Groups with the right permissions to interact with DynamoDB and assign it to the EC2 Instance.

7. You have created an AWS Lambda function that will write data to a DynamoDB table. Which of the following must be in place to ensure that the Lambda function can interact with the DynamoDB table?

A.
Ensure an IAM Role is attached to the Lambda function which has the required DynamoDB privileges.
B.
Ensure an IAM User is attached to the Lambda function which has the required DynamoDB privileges.
C.
Ensure the Access keys are embedded in the AWS Lambda function.
D.
Ensure the IAM user password is embedded in the AWS Lambda function.

8. You have both production and development-based instances running on your VPC. It is required to ensure that people responsible for the development instances do not have access to work on production instances for better security. Which of the following would be the best way to accomplish this using policy? Choose the correct answer from the options given below.

A.
Launch the test and production instances in separate VPCs and use VPC Peering.
B.
Create an IAM Policy with a condition that allows access to only those instances which are used for production or development.
C.
Launch the test and production instances in different Availability Zones and use Multi-Factor Authentication.

D.
Define the tags on the test and production servers and add a condition to the IAM Policy which allows access to specific tags.

9. One plans on using SQS queues and AWS Lambda to leverage the serverless aspects of the AWS Cloud. Each invocation to AWS Lambda will send a message to an SQS queue. For messages to be sent, which of the following must be in place?

A.
The queue must be a FIFO queue.
B.
An IAM Role with the required permissions.
C.
The code for Lambda must be written in C#.
D.
An IAM Group with the required permissions.

10. You work in the media industry and have created a web application where users will be able to upload photos they create to your website. This web application must be able to call the S3 API in order to be able to function. Where should you store your API credentials whilst maintaining the maximum level of security?

A.
Save the API credentials to your PHP files.
B.
Don't save your API credentials. Instead create a role in IAM and assign this role to an EC2 instance when you first create it.
C.
Save your API credentials in a public GitHub repository.
D.
Pass API credentials to the instance using instance user data.

11. Your company is planning on hosting their development, test and production applications on EC2 Instances in AWS. They are worried about how access control would be given to relevant IT Admins for each of the above environments. As an architect, what would you suggest for managing the relevant accesses?

A.
Add tags to the instances marking each environment and then segregate access using IAM Policies.
B.
Add User data to the underlying instances to mark each environment.
C.
Add Metadata to the underlying instances to mark each environment.
D.
Add each environment to a separate Auto Scaling Group.

12. An EC2 Instance setup in AWS will host an application which will make API calls to the Simple Storage Service. What is an ideal way for the application to access the Simple Storage Service?

A.
Pass API credentials to the instance using instance user data.
B.
Store API credentials as an object in a separate Amazon S3 bucket.
C.
Embed the API credentials into your application.
D.
Create and Assign an IAM role to the EC2 Instance.

13. You are developing a mobile application that needs to issue temporary security credentials to users. This is essential due to security concerns. Which of the below services can help achieve this?

A.
AWS STS
B.
AWS Config
C.
AWS Trusted Advisor
D.
AWS Inspector

14. Your company is planning on using the API Gateway service to manage APIs for developers and users. There is a need to segregate the access rights for both developers and users. How can this be accomplished?

A.
Use IAM permissions to control the access.
B.
Use AWS Access keys to manage the access.
C.
Use AWS KMS service to manage the access.
D.
Use AWS Config Service to control the access.

15. Your company has a set of EC2 Instances that access data objects stored in an S3 bucket. Your IT Security department is concerned about the security of this architecture and wants you to implement the following:
1) Ensure that the EC2 Instance securely accesses the data objects stored in the S3 bucket
2) Prevent accidental deletion of objects
Which of the following would help fulfill the requirements of the IT Security department? Choose 2 answers from the options given below.

A.
Create an IAM user and ensure the EC2 Instances use the IAM user credentials to access the data in the bucket.
B.
Create an IAM Role and ensure the EC2 Instances use the IAM Role to access the data in the bucket.
C.
Use S3 Cross-Region Replication to replicate the objects so that the integrity of data is maintained.
D.
Use an S3 bucket policy that ensures that MFA Delete is set on the objects in the bucket.

Answers:

1. **D**
   AWS Documentation mentions the following: With IAM roles for Amazon ECS tasks, you can specify an IAM role to be used by the containers in a task. Applications are required to sign their AWS API requests with AWS credentials, and this feature provides a strategy to manage credentials for your application's use. This is like how Amazon EC2 instance profiles provide credentials to EC2 instances.
2. **D**
   You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. It is not a good practice to use IAM credentials for a production-based application.

3. **C**
   STS generates Git Credentials for IAM users. The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS Identity and Access Management (IAM) users or for users that you authenticate (federated users)

4. **B**
   IAM User used to generate Federated User credentials does not have access on S3 bucket. Returns a set of temporary security credentials (consisting of an access key ID, a secret access key, and a security token) for a federated user. A typical use is in a proxy application that gets temporary security credentials on behalf of distributed applications inside a corporate network. You must call the GetFederationToken operation using the long-term security credentials of an IAM user. As a result, this call is appropriate in contexts where those credentials can be safely stored, usually in a server-based application.

5. **A**
   Returns a set of temporary security credentials for users who have been authenticated via a SAML authentication response. This operation provides a mechanism for tying an enterprise identity store or directory to role-based AWS access without user-specific credentials or configuration.

6. **A**
   An IAM role is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials (password or access keys) associated with it. Instead, if a user assumes a role, temporary security credentials are created dynamically and provided to the user. You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. Note: You can attach IAM role to the existing EC2 instance.

7. **A**
   AWS Documentation mentions the following to support this requirement.
   Each Lambda function has an IAM role (execution role) associated with it. You specify the IAM role when you create your Lambda function. Permissions you grant to this role determine what AWS Lambda can do when it assumes the role. There are two types of permissions that you grant to the IAM role:
   If your Lambda function code accesses other AWS resources, such as to read an object from an S3 bucket or write logs to CloudWatch Logs, you need to grant permissions for relevant Amazon S3 and CloudWatch actions to the role.
   If the event source is stream-based (Amazon Kinesis Data Streams and DynamoDB streams), AWS Lambda polls these streams on your behalf. AWS Lambda needs permissions to poll the stream and read new records on the stream so you need to grant the relevant permissions to this role.

8. **D**
9. **B**
   While working with AWS Lambda functions, if there is a need to access other resources, ensure that an IAM role is in place. The IAM role will have the required permissions to access the SQS queue
10. **B**
    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 behalf, 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.
11. **A**
    Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you've assigned to it. Each tag consists of a key and an optional value, both of which you define. For example, you could define a set of tags for your account's Amazon EC2 instances that helps you track each instance's owner and stack level. We recommend that you devise a set of tag keys that meets your needs for each resource type. Using a consistent set of tag keys makes it easier for you to manage your resources. You can search and filter the resources based on the tags you add.
12. **D**
    You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. It is not a good practice to use IAM credentials for a production-based application. It is always a good practice to use IAM Roles.
13. **A**

You can use the AWS Security Token Service (AWS STS) to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials are short-term, as the name implies. They can be configured to last for anywhere from a few minutes to several hours. After the credentials expire, AWS no longer recognizes them or allows any kind of access from API requests made with them.

14. **A**

To create, deploy, and manage an API in API Gateway, you must grant the API developer permissions to perform the required actions supported by the API management component of API Gateway.

To call a deployed API or to refresh the API caching, you must grant the API caller permissions to perform required IAM actions supported by the API execution component of API Gateway.

15. **B & D**

AWS Documentation mentions the following:

IAM roles are designed so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles.