**QUESTION 114**

A company is planning to run a group of Amazon EC2 instances that connect to an Amazon Aurora database.

The company has built an AWS CloudFormation template to deploy the EC2 instances and the Aurora DB cluster.

The company wants to allow the instances to authenticate to the database in a secure way.

The company does not want to maintain static database credentials.

Which solution meets these requirements with the LEAST operational effort?

1. Create a database user with a user name and password.

Add parameters for the database user name and password to the CloudFormation template.

Pass the parameters to the EC2 instances when the instances are launched.

1. Create a database user with a user name and password.

Store the user name and password in AWS Systems Manager Parameter Store.

Configure the EC2 instances to retrieve the database credentials from Parameter Store.

1. Configure the DB cluster to use IAM database authentication.

Create a database user to use with IAM authentication.

Associate a role with the EC2 instances to allow applications on the instances to access the database.

1. Configure the DB cluster to use IAM database authentication with an IAM user.

Create a database user that has a name that matches the IAM user.

Associate the IAM user with the EC2 instances to allow applications on the instances to access the database.

**Answer:** A

**Explanation:**

Finally, you need a way to instruct CloudFormation to complete stack creation only after all the services (such as Apache and MySQL) are running and not after all the stack resources are created. In other words, if you use the template from the earlier section to launch a stack, CloudFormation sets the status of the stack as CREATE\_COMPLETE after it successfully creates all the resources. However, if one or more services failed to start, CloudFormation still sets the stack status as CREATE\_COMPLETE. To prevent the status from changing to

CREATE\_COMPLETE until all the services have successfully started, you can add a

CreationPolicy attribute to the instance. This attribute puts the instance's status in

CREATE\_IN\_PROGRESS until CloudFormation receives the required number of success signals or the timeout period is exceeded, so you can control when the instance has been successfully created. Reference:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/deploying.applications.html>