

AWS Certified Developer – Associate (DVA-C02)

Exemples de questions pour l'examen

1) Une entreprise migre une application héritée vers des instances Amazon EC2. L'application se sert d'un nom d'utilisateur et d'un mot de passe stockés dans le code source pour se connecter à une base de données MySQL. L'entreprise migrera la base de données vers une instance DB Amazon RDS pour MySQL. Dans le cadre de la migration, l'entreprise doit mettre en place un moyen sécurisé de stocker et de faire tourner automatiquement les justificatifs de la base de données.

Quelle solution répondra à ces exigences ?

- A) Stocker les justificatifs de la base de données dans des variables d'environnement dans une Amazon Machine Image (AMI). Faire tourner les justificatifs en remplaçant l'AMI.
- B) Stocker les justificatifs de la base de données dans AWS Systems Manager Parameter Store. Configurer le Parameter Store pour faire tourner automatiquement les justificatifs.
- C) Stocker les justificatifs de la base de données dans des variables d'environnement sur les instances EC2. Faire tourner les justificatifs en relançant les instances EC2.
- D) Stocker les justificatifs de la base de données dans AWS Secrets Manager. Configurer Secrets Manager pour faire tourner automatiquement les justificatifs.

2) Un développeur crée une application web qui doit permettre aux utilisateurs de publier des commentaires et de recevoir un feedback en temps quasi réel.

Quelles solutions répondront à ces exigences ? (Sélectionnez DEUX réponses.)

- A) Créer un schéma AWS AppSync et les API correspondantes. Utiliser une table Amazon DynamoDB en tant que magasin de données.
- B) Créer une API WebSocket dans Amazon API Gateway. Utiliser une fonction AWS Lambda comme backend. Utiliser une table Amazon DynamoDB en tant que magasin de données.
- C) Créer une application AWS Elastic Beanstalk reposant sur une base de données Amazon RDS. Configurer l'application pour autoriser les sockets TCP/IP durables.
- D) Créer un point de terminaison GraphQL dans Amazon API Gateway. Utiliser une table Amazon DynamoDB en tant que magasin de données.
- E) Établir des connexions WebSocket vers Amazon CloudFront. Utiliser une fonction AWS Lambda comme origine de la distribution CloudFront. Utiliser un cluster de bases de données Amazon Aurora comme magasin de données.

3) Un développeur ajoute des fonctionnalités d'inscription et de connexion à une application. L'application doit effectuer un appel d'API vers une solution d'analytique personnalisée pour journaliser les événements de connexion des utilisateurs.

Quelle combinaison d'actions le développeur doit-il effectuer pour répondre à ces exigences ? (Sélectionnez DEUX réponses.)

- A) Utiliser Amazon Cognito pour fournir la fonctionnalité d'inscription et de connexion.
- B) Utiliser AWS Identity and Access Management (IAM) pour fournir les fonctionnalités d'inscription et de connexion.

AWS Certified Developer – Associate (DVA-C02)

Exemples de questions pour l'examen

- C) Configurer une règle AWS Config pour effectuer l'appel d'API lorsqu'un utilisateur est authentifié.
- D) Appeler une méthode Amazon API Gateway pour effectuer l'appel d'API lorsqu'un utilisateur est authentifié.
- E) Appeler une fonction AWS Lambda pour effectuer l'appel d'API lorsqu'un utilisateur est authentifié.

4) Une entreprise utilise Amazon API Gateway pour ses API REST dans un compte AWS. Un développeur ne souhaite autoriser que les utilisateurs IAM d'un autre compte AWS à accéder aux API.

Quelle combinaison de mesures le développeur doit-il prendre pour répondre à ces exigences ? (Sélectionnez DEUX réponses.)

- A) Créer une politique d'autorisation IAM. Joindre la politique à chaque utilisateur IAM. Définir le type d'autorisation de méthode pour les API sur AWS_IAM. Utiliser Signature Version 4 pour signer les demandes d'API.
- B) Créer un groupe d'utilisateurs Amazon Cognito. Ajouter chaque utilisateur IAM au groupe d'utilisateurs. Définir le type d'autorisation de méthode pour les API sur COGNITO_USER_POOLS. S'authentifier à l'aide des justificatifs IAM dans Amazon Cognito. Ajouter le jeton d'identification aux en-têtes de demande.
- C) Configurer un groupe d'identités Amazon Cognito. Ajouter chaque utilisateur IAM au groupe d'identités. Définir le type d'autorisation de méthode pour les API sur COGNITO_USER_POOLS. S'authentifier à l'aide des justificatifs IAM dans Amazon Cognito. Ajouter le jeton d'accès aux en-têtes de demande.
- D) Créer une politique de ressources pour les API qui n'autorise l'accès que pour chaque utilisateur IAM.
- E) Créer un mécanisme d'autorisation Amazon Cognito pour les API qui n'autorise l'accès que pour chaque utilisateur IAM. Définir le type d'autorisation de méthode pour les API sur COGNITO_USER_POOLS.

5) Un développeur crée une nouvelle application qui transforme des fichiers texte en fichiers .pdf. Une application distincte écrit les fichiers texte dans un compartiment Amazon S3 source. La nouvelle application doit lire les fichiers dès leur arrivée dans Amazon S3 et les convertir en fichiers .pdf à l'aide d'une fonction AWS Lambda. Le développeur a rédigé une politique IAM pour autoriser l'accès à Amazon S3 et Amazon CloudWatch Logs.

Que doit faire le développeur pour s'assurer que la fonction Lambda dispose des bonnes autorisations ?

- A) Créer un rôle d'exécution Lambda à l'aide d'AWS Identity and Access Management (IAM). Joindre la politique IAM au rôle. Attribuer le rôle d'exécution Lambda à la fonction Lambda.
- B) Créer un utilisateur d'exécution Lambda à l'aide d'AWS Identity and Access Management (IAM). Joindre la politique IAM à l'utilisateur. Attribuer l'utilisateur d'exécution Lambda à la fonction Lambda.
- C) Créer un rôle d'exécution Lambda à l'aide d'AWS Identity and Access Management (IAM). Joindre la politique IAM au rôle. Stocker le rôle IAM sous forme de variable d'environnement dans la fonction Lambda.

AWS Certified Developer – Associate (DVA-C02) Exemples de questions pour l'examen

- D) Créer un utilisateur d'exécution Lambda à l'aide d'AWS Identity and Access Management (IAM). Joindre la politique IAM à l'utilisateur. Stocker les justificatifs des utilisateurs IAM sous forme de variables d'environnement dans la fonction Lambda.

6) Un développeur travaille sur une application qui stocke des données hautement confidentielles dans une base de données. Le développeur doit utiliser AWS Key Management Service (AWS KMS) avec un chiffrement d'enveloppe pour protéger les données.

Comment le développeur doit-il configurer le chiffrement des données pour répondre à ces exigences ?

- A) Chiffrer les données à l'aide d'une clé KMS. Stocker les données chiffrées dans la base de données.
- B) Chiffrer les données à l'aide d'une clé de données générée. Stocker les données chiffrées dans la base de données.
- C) Chiffrer les données à l'aide d'une clé de données générée. Stocker les données chiffrées et l'ID de clé de données dans la base de données.
- D) Chiffrer les données à l'aide d'une clé de données générée. Stocker les données chiffrées et la clé de données chiffrée dans la base de données.

7) Un développeur ajoute Amazon ElastiCache for Memcached à l'application de stockage de registres existante d'une entreprise. Le développeur a décidé d'utiliser le chargement différé selon une analyse des schémas courants de gestion de registre.

Quel exemple de pseudo-code pourra correctement mettre en place le chargement différé ?

- A)

```
record_value = db.query("UPDATE Records SET Details = {1} WHERE ID == {0}",
                        record_key, record_value)
cache.set (record_key, record_value)
```
- B)

```
record_value = cache.get(record_key)
if (record_value == NULL)
    record_value = db.query("SELECT Details FROM Records WHERE ID == {0}",
                          record_key)
cache.set (record_key, record_value)
```
- C)

```
record_value = cache.get (record_key)
db.query("UPDATE Records SET Details = {1} WHERE ID == {0}", record_key,
        record_value)
```
- D)

```
record_value = db.query("SELECT Details FROM Records WHERE ID == {0}",
                        record_key)
if (record_value != NULL)
    cache.set (record_key, record_value)
```

8) Un développeur est en train de créer une application web qui utilise Amazon API Gateway. Le développeur souhaite maintenir différents environnements pour les charges de travail de développement

AWS Certified Developer – Associate (DVA-C02) Exemples de questions pour l'examen

(dev) et de production (prod). L'API reposera sur une fonction AWS Lambda avec deux alias : l'un pour le développement et l'autre pour la production.

Que doit faire le développeur pour maintenir ces environnements avec LE MOINS de configuration possible ?

- A) Créer une API REST pour chaque environnement. Intégrer les API aux alias de développement et de production correspondants de la fonction Lambda. Déployer les API vers leurs étapes respectives. Accéder aux API en utilisant les URL d'étape.
- B) Créer une seule API REST. Intégrer l'API à la fonction Lambda en utilisant une variable d'étape à la place d'un alias. Déployer l'API en deux étapes différentes : le développement et la production. Créer une variable d'étape dans chaque étape avec des alias différents comme valeurs. Accéder à l'API à l'aide des différentes URL d'étape.
- C) Créer une seule API REST. Intégrer l'API à l'alias dev de la fonction Lambda. Déployer l'API dans l'environnement de développement. Configurer un déploiement de version canary pour l'environnement de production dans lequel elle s'intégrera à l'alias de production Lambda.
- D) Créer une seule API REST. Intégrer l'API à l'alias prod de la fonction Lambda. Déployer l'API dans l'environnement de production. Configurer un déploiement de version canary pour l'environnement de développement dans lequel elle s'intégrera à l'alias de développement Lambda.

9) Un développeur souhaite suivre les performances d'une application exécutée sur une flotte d'instances Amazon EC2. Le développeur souhaite afficher et suivre les statistiques, telles que la latence moyenne des demandes et la latence maximale des demandes, sur l'ensemble de la flotte. Le développeur souhaite recevoir une notification immédiate si le temps de réponse moyen dépasse un certain seuil.

Quelle solution répondra à ces exigences ?

- A) Configurer une tâche cron sur chaque instance EC2 afin de mesurer le temps de réponse et mettre à jour un fichier journal stocké dans un compartiment Amazon S3 toutes les minutes. Utiliser une notification d'événement Amazon S3 pour invoquer une fonction AWS Lambda qui lit le fichier journal et écrit de nouvelles entrées vers un cluster Amazon OpenSearch Service. Visualiser les résultats dans les tableaux de bord OpenSearch. Configurer OpenSearch Service pour envoyer une alerte à une rubrique Amazon Simple Notification Service (Amazon SNS) lorsque le temps de réponse dépasse le seuil.
- B) Configurer l'application pour qu'elle inscrive les temps de réponse dans le journal système. Installer et configurer l'agent Amazon Inspector sur les instances EC2 pour lire en continu les journaux et envoyer les temps de réponse à Amazon EventBridge (Amazon CloudWatch Events). Consulter les graphiques des métriques dans la console EventBridge (CloudWatch Events). Configurer une règle personnalisée EventBridge (CloudWatch Events) pour envoyer une notification Amazon Simple Notification Service (Amazon SNS) lorsque la moyenne de la métrique du temps de réponse dépasse le seuil.

AWS Certified Developer – Associate (DVA-C02)

Exemples de questions pour l'examen

- C) Configurer l'application pour écrire les temps de réponse dans un fichier journal. Installer et configurer l'agent Amazon CloudWatch sur les instances EC2 pour diffuser le journal des applications vers CloudWatch Logs. Créer un filtre des métriques du temps de réponse à partir du journal. Afficher les graphiques de métriques dans la console CloudWatch. Créer une alarme CloudWatch pour envoyer une notification Amazon Simple Notification Service (Amazon SNS) lorsque la moyenne de la métrique du temps de réponse dépasse le seuil.
- D) Installer et configurer AWS Systems Manager Agent (SSM Agent) sur les instances EC2 pour surveiller le temps de réponse et l'envoyer vers Amazon CloudWatch sous forme de métrique personnalisée. Afficher les graphiques de métriques dans Amazon QuickSight. Créer une alarme CloudWatch pour envoyer une notification Amazon Simple Notification Service (Amazon SNS) lorsque la moyenne de la métrique du temps de réponse dépasse le seuil.

10) Un développeur teste une application localement et la déploie sur une fonction AWS Lambda. Pour éviter de dépasser le quota de taille du package de déploiement, le développeur n'a pas inclus les dépendances dans le fichier de déploiement. Lorsque le développeur teste l'application à distance, la fonction Lambda ne s'exécute pas en raison des dépendances manquantes.

Quelle solution permettra de résoudre ce problème ?

- A) Utiliser l'éditeur de console Lambda pour mettre à jour le code et inclure les dépendances manquantes.
- B) Créer un fichier .zip supplémentaire contenant les dépendances manquantes. Inclure le fichier .zip dans le package de déploiement Lambda d'origine.
- C) Ajouter des références aux dépendances manquantes dans les variables d'environnement de la fonction Lambda.
- D) Créer une couche contenant les dépendances manquantes. Joindre la couche à la fonction Lambda.

AWS Certified Developer – Associate (DVA-C02)

Exemples de questions pour l'examen

Réponses

1) D – [AWS Secrets Manager](#) aide à protéger les justificatifs nécessaires pour accéder aux bases de données, applications, services et autres ressources informatiques. Avec Secrets Manager, vous pouvez faire tourner, gérer et récupérer les justificatifs d'une base de données, des clés d'API et d'autres secrets tout au long de leur cycle de vie. Les utilisateurs et les applications récupèrent des secrets grâce à un appel d'API Secrets Manager, éliminant ainsi le besoin de coder en dur des informations sensibles en texte brut. Secrets Manager propose la [rotation des secrets](#) grâce à l'intégration incorporée à Amazon RDS, Amazon Redshift et Amazon DocumentDB (compatible avec MongoDB).

2) A, B – [AWS AppSync](#) simplifie le développement d'applications en vous permettant de créer une API flexible afin d'accéder, de manipuler et de combiner en toute sécurité les données provenant d'une ou de plusieurs sources de données. AWS AppSync est un service géré qui utilise GraphQL pour aider les applications à extraire les données dont elles ont besoin. Vous pouvez utiliser AWS AppSync pour créer des applications pouvant être mises à l'échelle qui nécessitent des [mises à jour en temps réel](#) sur diverses sources de données, y compris Amazon DynamoDB.

Dans [Amazon API Gateway](#), vous pouvez [créer une API WebSocket](#) en tant que frontend avec état pour un service AWS (tel qu'AWS Lambda ou DynamoDB) ou pour un point de terminaison HTTP. L'API WebSocket appelle le backend en fonction du contenu des messages que l'API reçoit de la part des applications clientes. Contrairement à une API REST, qui reçoit des demandes et y répond, une API WebSocket prend en charge la communication bidirectionnelle entre les applications clientes et le backend.

3) A, E – [Amazon Cognito](#) ajoute l'inscription des utilisateurs, la connexion et le contrôle d'accès aux applications web et mobiles. Vous pouvez également créer une fonction AWS Lambda pour effectuer un appel d'API vers une solution d'analytique personnalisée, puis invoquer cette fonction à l'aide d'un [déclencheur post-authentification Amazon Cognito](#).

4) A, D – Une [politique de ressources](#) peut accorder l'accès API dans un compte AWS à des utilisateurs d'un compte AWS différent à l'aide des protocoles [Signature Version 4](#) (SigV4).

5) A – Un [rôle d'exécution](#) de fonction AWS Lambda accorde à une fonction Lambda l'autorisation d'accéder aux services et aux ressources AWS. Vous fournissez ce rôle lorsque vous créez une fonction et Lambda endosse le rôle lorsqu'une fonction est invoquée.

6) D – Le [chiffrement d'enveloppe](#) est la pratique qui consiste à chiffrer des données en texte brut avec une clé de données, puis à chiffrer la clé de données sous une autre clé. Vous devez stocker la forme chiffrée de la clé de données afin de pouvoir utiliser la clé de données pour déchiffrer les données chiffrées dans la base de données.

7) B – Le [chargement différé](#) est une stratégie de mise en cache dans laquelle un registre ne se charge pas tant que cela n'est pas nécessaire. Lorsque vous mettez en place le chargement différé, l'application vérifie d'abord la présence d'un registre dans le cache. Si le registre n'est pas présent, l'application récupère le registre de la base de données et le stocke dans le cache.

AWS Certified Developer – Associate (DVA-C02)

Exemples de questions pour l'examen

- 8) B – Avec les étapes de déploiement dans Amazon API Gateway, vous pouvez gérer plusieurs étapes de publication pour chaque API. Vous pouvez configurer des [variables d'étapes](#) afin qu'une étape de déploiement d'API puisse interagir avec différents points de terminaison backend. Vous pouvez utiliser des variables d'étape API Gateway pour [référencer une seule fonction AWS Lambda](#) avec plusieurs versions et alias.
- 9) C – Vous pouvez configurer [l'agent Amazon CloudWatch](#) pour diffuser des journaux et des métriques sur CloudWatch. Vous pouvez également créer des [filtres de métriques](#) à partir des journaux stockés dans CloudWatch Logs.
- 10) D – Vous pouvez configurer une fonction AWS Lambda pour extraire du code et du contenu supplémentaires sous forme de [couches](#). Une couche est une archive .zip qui contient des bibliothèques, une exécution personnalisée ou d'autres dépendances. Avec les couches, vous pouvez utiliser des bibliothèques dans une fonction Lambda sans avoir besoin de les inclure dans un package de déploiement.