1) Un'azienda sta eseguendo la migrazione di un'applicazione legacy verso istanze di Amazon EC2. L'applicazione usa un nome utente e una password archiviati nel codice sorgente per connettersi a un database MySQL. L'azienda eseguirà la migrazione del database verso un'istanza database Amazon RDS for MySQL. Nell'ambito della migrazione, l'organizzazione desidera implementare un modo sicuro per archiviare e ruotare automaticamente le credenziali del database.

Quale soluzione soddisfa questi requisiti?

A) Archiviare le credenziali di database nelle variabili di ambiente in una Amazon Machine Image (AMI). Ruotare le credenziali sostituendo l'AMI.
B) Archiviare le credenziali di database in AWS Systems Manager Parameter Store. Configurare il Parameter Store in modo che ruoti automaticamente le credenziali.
C) Archiviare le credenziali del database nelle variabili di ambiente nelle istanze EC2. Ruotare le credenziali lanciando nuovamente le istanze EC2.
D) Archiviare le credenziali del database in AWS Secrets Manager. Configurare Secrets Manager in modo che ruoti automaticamente le credenziali.

2) Uno sviluppatore sta creando un'applicazione web che deve dare agli utenti la possibilità di pubblicare commenti e ricevere feedback quasi in tempo reale.

Quali soluzioni soddisfano questi requisiti? (Selezionare DUE risposte.)

A) Creare uno schema AWS AppSync e le API corrispondenti. Usare una tabella Amazon DynamoDB come archivio dati.
B) Creare un'API WebSocket in Amazon API Gateway. Usare una funzione AWS Lambda come back-end. Usare una tabella Amazon DynamoDB come archivio dati.
C) Creare un'applicazione AWS Elastic Beanstalk supportata da un database Amazon RDS. Configurare l'applicazione per consentire socket TCP/IP di lunga durata.
D) Creare un endpoint GraphQL in Amazon API Gateway. Usare una tabella Amazon DynamoDB come archivio dati.
E) Stabilire connessioni WebSocket ad Amazon CloudFront. Usare una funzione AWS Lambda come origine della distribuzione CloudFront. Usare un cluster Amazon Aurora DB come archivio dati.

3) Uno sviluppatore sta aggiungendo la funzionalità di registrazione e accesso a un'applicazione. L'applicazione deve effettuare una chiamata API a una soluzione di analisi personalizzata per registrare gli eventi di accesso utente.

Quale combinazione di azioni deve eseguire lo sviluppatore per soddisfare questi requisiti? (Selezionare DUE risposte.)

A) Usare Amazon Cognito per fornire la funzionalità di registrazione e accesso.
B) Usare AWS Identity and Access Management (IAM) per fornire la funzionalità di registrazione e accesso.
C) Configurare una regola AWS Config per eseguire la chiamata API quando un utente viene autenticato.

D) Richiamare un metodo Amazon API Gateway per eseguire la chiamata API quando un utente viene autenticato.
E) Richiamare una funzione AWS Lambda per eseguire la chiamata API quando un utente viene autenticato.

4) Un'azienda usa Amazon API Gateway per le API REST in un account AWS. Uno sviluppatore desidera consentire solo agli utenti IAM di un altro account AWS di accedere alle API.

Quale combinazione di passaggi deve eseguire lo sviluppatore per soddisfare questi requisiti? (Selezionare DUE risposte.)

A) Creare una policy di autorizzazione IAM. Collegare la policy a ogni utente IAM. Impostare il tipo di autorizzazione del metodo per le API su AWS_IAM. Usare Signature Version 4 per firmare le richieste API.
B) Creare un pool di utenti Amazon Cognito. Aggiungere ogni utente IAM al pool di utenti. Impostare il tipo di autorizzazione del metodo per le API su COGNITO_USER_POOLS. Eseguire l'autenticazione tramite le credenziali IAM in Amazon Cognito. Aggiungere il token ID alle intestazioni della richiesta.
C) Configurare un pool di identità Amazon Cognito. Aggiungere ogni utente IAM al pool di identità. Impostare il tipo di autorizzazione del metodo per le API su COGNITO_USER_POOLS. Eseguire l'autenticazione tramite le credenziali IAM in Amazon Cognito. Aggiungere il token di accesso alle intestazioni della richiesta.
D) Creare una policy delle risorse per le API per consentire l'accesso solo a ciascun utente IAM.
E) Creare un provider di autorizzazioni Amazon Cognito per le API per consentire l'accesso solo a ciascun utente IAM. Impostare il tipo di autorizzazione del metodo per le API su COGNITO_USER_POOLS.

5) Uno sviluppatore sta creando una nuova applicazione che trasforma i file di testo in file .pdf. Un'applicazione separata scrive i file di testo su un bucket Amazon S3 di origine. La nuova applicazione deve leggere i file non appena arrivano in Amazon S3 e convertirli in file .pdf tramite una funzione AWS Lambda. Lo sviluppatore ha scritto una policy IAM per consentire l'accesso ad Amazon S3 e ad Amazon CloudWatch Logs.

Quali azioni deve eseguire lo sviluppatore per garantire che la funzione Lambda disponga delle autorizzazioni corrette?

A) Creare un ruolo di esecuzione Lambda tramite AWS Identity and Access Management (IAM). Allegare la policy IAM al ruolo. Assegnare il ruolo di esecuzione Lambda alla funzione Lambda.
B) Creare un utente di esecuzione Lambda tramite AWS Identity and Access Management (IAM). Allegare la policy IAM all'utente. Assegnare l'utente di esecuzione Lambda alla funzione Lambda.
C) Creare un ruolo di esecuzione Lambda tramite AWS Identity and Access Management (IAM). Allegare la policy IAM al ruolo. Archiviare il ruolo IAM come variabile di ambiente nella funzione Lambda.
D) Creare un utente di esecuzione Lambda tramite AWS Identity and Access Management (IAM). Allegare la policy IAM all'utente. Archiviare le credenziali dell'utente IAM come variabili di ambiente nella funzione Lambda.

6) Uno sviluppatore sta lavorando a un'applicazione che archivia dati altamente riservati in un database. Lo sviluppatore deve usare AWS Key Management Service (AWS KMS) con crittografia envelope per proteggere i dati.

Come deve procedere lo sviluppatore per configurare la crittografia dei dati in modo che soddisfi questi requisiti?

A) Crittografare i dati usando una chiave KMS. Archiviare i dati crittografati nel database.
B) Crittografare i dati usando una chiave dati generata. Archiviare i dati crittografati nel database.
C) Crittografare i dati usando una chiave dati generata. Archiviare i dati crittografati e l'ID della chiave dati nel database.
D) Crittografare i dati usando una chiave dati generata. Archiviare i dati crittografati e la chiave dati crittografata nel database.

7) Uno sviluppatore sta aggiungendo Amazon ElastiCache for Memcached all'applicazione di archiviazione di record esistente di un'azienda. Lo sviluppatore ha deciso di usare il caricamento lazy sulla base di un'analisi dei modelli comuni di gestione dei record.

Quale esempio di pseudocodice implementerà correttamente il caricamento lazy?

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) Uno sviluppatore sta creando un'applicazione web che usa Amazon API Gateway. Lo sviluppatore desidera gestire ambienti diversi per i carichi di lavoro di sviluppo (dev) e produzione (prod). L'API sarà supportata da una funzione AWS Lambda con due alias: uno per l'ambiente di sviluppo e uno per quello di produzione.

Come deve procedere lo sviluppatore per gestire questi ambienti con la MINOR quantità di configurazione?

A) Creare un'API REST per ogni ambiente. Integrare le API con i corrispondenti alias dev e prod della funzione Lambda. Distribuire le API nelle rispettive fasi. Accedere alle API tramite gli URL della fase.
B) Creare un'unica API REST. Integrare l'API con la funzione Lambda tramite una variabile di fase al posto di un alias. Distribuire l'API in due diverse fasi: dev e prod. Creare una variabile di fase in ogni fase con alias diversi come valori. Accedere all'API tramite gli URL delle diverse fasi.
C) Creare un'unica API REST. Integrare l'API con l'alias dev della funzione Lambda. Distribuire l'API nell'ambiente dev. Configurare la distribuzione di una canary release per l'ambiente prod in cui la canary si integrerà con l'alias prod di Lambda.
D) Creare un'unica API REST. Integrare l'API con l'alias prod della funzione Lambda. Distribuire l'API nell'ambiente prod. Configurare la distribuzione di una canary release per l'ambiente dev in cui la canary si integrerà con l'alias dev di Lambda.

9) Uno sviluppatore desidera monitorare le prestazioni di un'applicazione che viene eseguita su un parco di istanze Amazon EC2. Lo sviluppatore desidera visualizzare e monitorare le statistiche, come la latenza media e massima delle richieste, per tutto il parco di istanze e desidera ricevere una notifica immediata se il tempo di risposta medio supera una determinata soglia.

Quale soluzione soddisfa questi requisiti?

A) Configurare un processo cron su ogni istanza EC2 per misurare il tempo di risposta e aggiornare ogni minuto un file di log archiviato in un bucket Amazon S3. Usare una notifica di evento Amazon S3 per richiamare una funzione AWS Lambda in grado di leggere il file di log e scrivere nuove voci in un cluster Amazon OpenSearch Service. Visualizzare i risultati nella dashboard di OpenSearch. Configurare OpenSearch Service per inviare un avviso a un argomento Amazon Simple Notification Service (Amazon SNS) quando il tempo di risposta supera la soglia.
B) Configurare l'applicazione affinché scriva i tempi di risposta nel log di sistema. Installare e configurare l'agente Amazon Inspector sulle istanze EC2 per leggere continuamente i log e inviare i tempi di risposta ad Amazon EventBridge (Amazon CloudWatch Events). Visualizzare i grafici delle metriche nella console EventBridge (Amazon CloudWatch Events). Configurare una regola personalizzata EventBridge (Amazon CloudWatch Events) per inviare una notifica Amazon SNS quando la media della metrica relativa al tempo di risposta supera la soglia.
C) Configurare l'applicazione affinché scriva i tempi di risposta in un file di log. Installare l'agente Amazon CloudWatch sulle istanze EC2 e configurarlo affinché trasmetta il log dell'applicazione a CloudWatch Logs. Creare un filtro per la metrica del tempo di risposta dal log. Visualizzare i grafici della metrica nella console CloudWatch. Creare un allarme CloudWatch per inviare una notifica Amazon Simple Notification Service (Amazon SNS) quando la media della metrica relativa al tempo di risposta supera la soglia.
D) Installare l'agente AWS Systems Manager (agente SSM) sulle istanze EC2 e configurarlo per monitorare e inviare il tempo di risposta ad Amazon CloudWatch sotto forma di metrica personalizzata. Visualizzare i grafici della metrica in Amazon QuickSight. Creare un allarme CloudWatch per inviare una notifica Amazon Simple Notification Service (Amazon SNS) quando la media della metrica relativa al tempo di risposta supera la soglia.

10) Uno sviluppatore sta testando un'applicazione in locale e ha distribuito l'applicazione su una funzione AWS Lambda. Per evitare di superare la quota relativa alla dimensione del pacchetto di distribuzione, lo sviluppatore non ha incluso le dipendenze nel file di distribuzione. Quando testa l'applicazione in remoto, la funzione Lambda non viene eseguita a causa di dipendenze mancanti.

Quale soluzione risolverà il problema?

A) Usare l'editor della console Lambda per aggiornare il codice e includere le dipendenze mancanti.
B) Creare un file .zip aggiuntivo contenente le dipendenze mancanti. Includere il file .zip nel pacchetto di distribuzione Lambda originale.
C) Aggiungere i riferimenti alle dipendenze mancanti nelle variabili di ambiente della funzione Lambda.
D) Creare un layer che contenga le dipendenze mancanti. Collegare il layer alla funzione Lambda.

Risposte

1) D. AWS Secrets Manager consente di proteggere le credenziali necessarie per l'accesso a database, applicazioni, servizi e altre risorse IT. Con Secrets Manager è possibile ruotare, gestire e recuperare facilmente credenziali del database, chiavi API e altri segreti durante il loro ciclo di vita. Gli utenti e le applicazioni recuperano i segreti effettuando una chiamata alle API di Secrets Manager; di conseguenza, non è più necessario codificare le informazioni sensibili in un testo in chiaro. Secrets Manager offre una rotazione segreta con integrazione predefinita per Amazon RDS, Amazon Redshift e Amazon DocumentDB (compatibile con MongoDB).

2) A, B. AWS AppSync semplifica lo sviluppo delle applicazioni consentendo di creare un'API flessibile per accedere, manipolare e combinare in modo sicuro i dati provenienti da una o più origini dati. AWS AppSync è un servizio gestito che usa GraphQL per consentire alle applicazioni di ottenere proprio i dati di cui hanno bisogno. AWS AppSync consente di creare applicazioni scalabili che richiedono aggiornamenti in tempo reale su una vasta gamma di origini dati, tra cui Amazon DynamoDB.

In Amazon API Gateway puoi creare un'API WebSocket come front-end stateful per un servizio AWS (ad esempio AWS Lambda o DynamoDB) o per un endpoint HTTP. L'API WebSocket richiama il back-end in base al contenuto dei messaggi che l'API riceve dalle applicazioni client. A differenza di un'API REST, che riceve e risponde alle richieste, un'API WebSocket supporta la comunicazione bidirezionale tra le applicazioni client e il back-end.

3) A, E. Amazon Cognito consente di aggiungere la registrazione, l'accesso e il controllo degli accessi alle applicazioni web e per dispositivi mobili. È inoltre possibile creare una funzione AWS Lambda per effettuare una chiamata API a una soluzione di analisi personalizzata e quindi richiamare tale funzione con un trigger post autenticazione di Amazon Cognito.

4) A, D. Una policy delle risorse può concedere l'accesso API in un account AWS agli utenti di un account AWS diverso usando i protocolli Signature Version 4 (SigV4).

5) A. Un ruolo di esecuzione della funzione AWS Lambda concede alla funzione Lambda l'autorizzazione per accedere ai servizi e alle risorse AWS. Questo ruolo viene fornito quando si crea una funzione e Lambda assume tale ruolo quando viene richiamata una funzione.

6) D. La crittografia envelope consiste nel crittografare i dati in un testo in chiaro con una chiave dati e quindi crittografare a sua volta la chiave dati con un'altra chiave. È necessario archiviare il formato crittografato della chiave dati in modo da poterla usare per decrittografare i dati crittografati nel database.

7) B. Il caricamento lazy è una strategia di memorizzazione nella cache in cui un record non viene caricato finché non è richiesto. Con l'implementazione del caricamento lazy, l'applicazione controlla innanzitutto se nella cache è presente un record. Se non è presente, l'applicazione lo recupera dal database e lo memorizza nella cache.

8) B. Le fasi di distribuzione in Amazon API Gateway consentono di gestire più fasi di rilascio per ciascuna API. È possibile configurare variabili di fase perché una fase di distribuzione API possa interagire con diversi endpoint di back-end. Le variabili di fase API Gateway permettono di fare riferimento a una singola funzione AWS Lambda con più versioni e alias.

9) C. È possibile configurare l'agente Amazon CloudWatch per trasmettere log e metriche a CloudWatch. È inoltre possibile creare filtri per le metriche dai log archiviati in CloudWatch Logs.

10) D. È possibile configurare una funzione AWS Lambda per inserire codice e contenuti aggiuntivi sotto forma di layer. Un layer è un archivio di file .zip che contiene librerie, un runtime personalizzato o altre dipendenze. Con i layer, è possibile usare le librerie in una funzione Lambda senza dover includere le librerie in un pacchetto di distribuzione.