

501 (신재경,김지형) 5회차 完

◆ | Q#0501. | Ref#0501.

한 회사가 단일 Amazon EC2 인스턴스에서 웹 애플리케이션을 실행합니다. 최종 사용자는 CPU 사용률이 지속적으로 95%를 초과하는 피크 사용 시간에 애플리케이션 성능이 느려지는 것을 경험합니다.

사용자 데이터 스크립트는 EC2 인스턴스에 필요한 사용자 지정 패키지를 설치합니다. 인스턴스를 시작하는 프로세스에는 몇 분이 걸립니다.

이 회사는 혼합 인스턴스 그룹, 다양한 CPU 및 최대 용량 제한이 있는 자동 확장 그룹을 만들고 있습니다. 자동 확장 그룹은 다양한 구성 옵션에 대한 시작 템플릿을 사용합니다. 이 회사는 자동 확장 중에 새 인스턴스가 시작될 때 애플리케이션 대기 시간을 줄여야 합니다.

어떤 솔루션이 이러한 요구 사항을 충족할까요?

- A.** 예측적 확장 정책을 사용합니다. 인스턴스 유지 관리 정책을 사용하여 사용자 데이터 스크립트를 실행합니다. 기본 인스턴스 워밍업 시간을 0초로 설정합니다.
- B.** 동적 스케일링 정책을 사용합니다. 라이프사이클 후크를 사용하여 사용자 데이터 스크립트를 실행합니다. 기본 인스턴스 워밍업 시간을 0초로 설정합니다.
- C.** 예측적 스케일링 정책을 사용합니다. 자동 스케일링 그룹에 대한 워밍업을 활성화합니다. 인스턴스 유지 관리 정책을 사용하여 사용자 데이터 스크립트를 실행합니다.
- D.** 동적 스케일링 정책을 사용합니다. 자동 스케일링 그룹에 대한 워밍업을 활성화합니다. 라이프사이클 후크를 사용하여 사용자 데이터 스크립트를 실행합니다.

해설

정답: D

동적 스케일링 정책과 워밍업을 사용하면 트래픽 변화에 유연하게 대응하면서 새 인스턴스의 응답 시간을 최소화할 수 있습니다. 또한, 라이프사이클 후크를 통해 사용자 데이터 스크립트를 인스턴스가 완전히 시작된 후 실행할 수 있으므로 인스턴스의 설정을 제대로 완료할 수 있습니다.

동적 스케일링 정책은 현재 트래픽에 따라 스케일 조정을 수행합니다. 이는 실시간 트래픽에 대응할 수 있으므로 유용합니다.

워밍업을 사용하면 새 인스턴스가 빠르게 준비되므로 응답 시간을 줄일 수 있습니다.

라이프사이클 후크를 사용하여 인스턴스가 시작될 때 사용자 데이터 스크립트를 실행할 수 있습니다.

이는 인스턴스가 완전히 준비된 후 필요한 설정을 완료하는 데 유용합니다.

◆ | Q#0502. | Ref#0502.

한 회사가 온프레미스 데이터베이스 플릿을 Amazon RDS로 마이그레이션해야 합니다. 이 회사는 현재 Microsoft SQL Server, MySQL, Oracle 데이터베이스를 혼합하여 사용하고 있습니다. 일부 데이터베이스에는 사용자 지정 스키마와 저장 프로시저가 있습니다.

이 회사는 마이그레이션을 위해 어떤 단계 조합을 취해야 할까요? (두 가지를 선택하세요.)

- A.** Migration Evaluator Quick Insights를 사용하여 소스 데이터베이스를 분석하고 마이그레이션이 필요한 저장 프로시저를 식별합니다.
- B.** AWS Application Migration Service를 사용하여 소스 데이터베이스를 분석하고 마이그레이션이 필요한 저장 프로시저를 식별합니다.
- C.** AWS Schema Conversion Tool(AWS SCT)을 사용하여 필요한 변경 사항을 위해 소스 데이터베이스를 분석합니다.
- D.** AWS Database Migration Service(AWS DMS)를 사용하여 소스 데이터베이스를 Amazon RDS로 마이그레이션합니다.
- E.** AWS DataSync를 사용하여 소스 데이터베이스의 데이터를 Amazon RDS로 마이그레이션합니다.

해설

정답: C,D

마이그레이션 시 스키마 및 저장 프로시저의 변환을 지원하는 도구와 데이터 이동을 지원하는 도구를 조합하여 사용하는 것이 이상적입니다.

옵션 C: AWS Schema Conversion Tool (AWS SCT)는 데이터베이스 스키마 및 저장 프로시저의 변환을 지원합니다. 이 도구는 다양한 데이터베이스 엔진 간의 스키마와 저장 프로시저를 분석하고 변환하는 데 도움을 줍니다. 따라서 이 옵션은 적절합니다.

옵션 D: AWS Database Migration Service (AWS DMS)는 데이터베이스의 데이터를 클라우드로 이동하는 데 사용됩니다. 데이터베이스의 데이터는 효율적으로 마이그레이션할 수 있지만, 데이터베이스의 스키마나 저장 프로시저의 변환을 지원하지는 않습니다. 따라서 이 옵션은 데이터 이동에는 적합하지만 스키마 변환에는 적합하지 않습니다.

◆ | Q#0503. | Ref#0503.

한 회사가 블로그 플랫폼을 AWS로 마이그레이션하고 있습니다. 회사의 온프레미스 서버는 AWS 사이트 간 VPN 연결을 통해 AWS에 연결됩니다. 블로그 콘텐츠는 여러 작성자가 하루에 여러 번 업데이트하고 NAS(Network Attached Storage) 서버의 파일 공유에서 제공됩니다.

회사는 콘텐츠 업데이트를 지연시키지 않고 블로그 플랫폼을 마이그레이션해야 합니다. 회사는 여러 가용성 영역에 Amazon EC2 인스턴스를 배포하여 애플리케이션 로드 밸런서 뒤에서 블로그 플랫폼을 실행했습니다. 또한 회사는 가능한 한 빨리 온프레미스 서버에서 Amazon S3로 200TB의 보관 데이터를 이동해야 합니다.

어떤 정거장 조합이 이러한 요구 사항을 충족할까요? (두 가지를 선택하세요.)

A. Amazon EventBridge에서 주간 cron 작업을 만듭니다. cron 작업을 사용하여 AWS Lambda 함수를 호출하여 NAS 서버에서 EC2 인스턴스를 업데이트합니다.

B. EC2 인스턴스가 콘텐츠 액세스를 위해 공유할 Amazon Elastic Block Store(Amazon EBS) Multi-Attach 볼륨을 구성합니다. EBS 볼륨을 NAS 서버와 매주 동기화하는 코드를 작성합니다.

C. 온프레미스 서버에 Amazon Elastic File System(Amazon EFS) 파일 시스템을 마운트하여 NAS 서버로 작동시킵니다. 블로그 데이터를 EFS 파일 시스템에 복사합니다. EFS 파일 시스템을 EC2 인스턴스에 마운트하여 콘텐츠를 제공합니다.

D. AWS Snowball Edge Storage Optimized 장치를 주문합니다. 정적 데이터 아티팩트를 장치에 복사합니다. 장치를 AWS로 배송합니다.

E. AWS Snowcons SSD 장치를 주문합니다. 정적 데이터 아티팩트를 장치에 복사합니다. 장치를 AWS로 배송합니다.

해설

정답: C,D

블로그 플랫폼의 콘텐츠 업데이트를 지연 없이 처리하려면 Amazon EFS를 사용하는 것이 적합하며, 200TB의 데이터를 신속하게 이동하려면 AWS Snowball Edge를 사용하는 것이 좋습니다.

옵션 C: Amazon EFS는 파일 시스템을 EC2 인스턴스와 온프레미스 서버에서 동시에 사용할 수 있도록 해줍니다. EFS를 온프레미스 서버와 EC2 인스턴스에서 모두 마운트하면 실시간 콘텐츠 업데이트가 가능하고, 블로그 데이터를 EFS로 복사하여 EC2 인스턴스에서 제공할 수 있습니다. 이 방법은 콘텐츠 업데이트 지연 문제를 해결합니다.

옵션 D: AWS Snowball Edge Storage Optimized는 대량의 데이터를 오프라인으로 이동할 때 유용합니다. 데이터가 200TB인 경우, Snowball Edge는 좋은 선택이 될 수 있습니다. 하지만, 이 방법은 실시간 콘텐츠 업데이트에는 관련이 없습니다. 아카이브 데이터 이동에는 적합합니다.

◆ | Q#0504. | Ref#0504.

한 회사가 레거시 온프레미스 애플리케이션을 AWS로 마이그레이션할 계획입니다. 이 애플리케이션은 PostgreSQL 데이터베이스가 있는 Apache Tomcat에서 실행되는 Java 웹 애플리케이션입니다.

이 회사는 소스 코드에 액세스할 수 없지만 애플리케이션 Java Archive(JAR) 파일을 배포할 수 있습니다. 이 애플리케이션은 매월 말에 트래픽이 증가합니다.

어떤 솔루션이 이러한 요구 사항을 가장 적은 운영 오버헤드로 충족할까요?

- A.** 여러 가용 영역에서 Amazon EC2 인스턴스를 시작합니다. Amazon Elastic File System(Amazon EFS) 마운트 포인트를 사용하여 모든 인스턴스에 Tomcat과 PostgreSQL을 배포합니다. AWS Step Functions를 사용하여 추가 EC2 인스턴스를 배포하여 트래픽 증가에 맞게 확장합니다.
- B.** 여러 AWS 리전에서 Auto Scaling 그룹으로 Amazon Elastic Kubernetes Service(Amazon EKS)를 프로비저닝합니다. 컨테이너 이미지에 Tomcat과 PostgreSQL을 배포합니다. 네트워크 로드 밸런서를 사용하여 트래픽 증가에 맞게 확장합니다.
- C.** Java 애플리케이션을 Python 기반 컨테이너로 리팩토링합니다. 애플리케이션 로직에 AWS Lambda 함수를 사용합니다. 애플리케이션 데이터를 Amazon DynamoDB 글로벌 테이블에 저장합니다. AWS Storage Gateway와 Lambda 동시성을 사용하여 트래픽 증가에 맞게 확장합니다.
- D.** AWS Elastic Beanstalk를 사용하여 여러 가용성 영역에 자동 스케일링이 있는 Tomcat 서버를 배포합니다. Amazon RDS for PostgreSQL 데이터베이스에 애플리케이션 데이터를 저장합니다. Amazon CloudFront와 Application Load Balancer를 배포하여 트래픽 증가에 맞게 확장합니다.

해설

정답: D

현재 애플리케이션을 최소한의 운영 오버헤드로 AWS에 마이그레이션하려면, 애플리케이션의 관리와 스케일링을 간소화할 수 있는 솔루션을 선택하는 것이 중요합니다. AWS Elastic Beanstalk와 Amazon RDS for PostgreSQL, 그리고 Application Load Balancer와 Amazon CloudFront를 조합하면 최소한의 관리 오버헤드로 스케일링을 효과적으로 처리할 수 있습니다.

AWS Elastic Beanstalk는 애플리케이션 배포와 관리를 간소화해주는 PaaS(Platform as a Service)입니다. Tomcat 서버를 자동으로 배포하고 관리할 수 있습니다.

Amazon RDS for PostgreSQL은 관리형 데이터베이스 서비스로, PostgreSQL 데이터베이스를 쉽게 운영할 수 있습니다.

Application Load Balancer는 HTTP 트래픽을 효과적으로 처리하며, Amazon CloudFront를 사용하여 콘텐츠를 전 세계적으로 캐싱하고 전달할 수 있습니다.

Elastic Beanstalk의 자동 스케일링 기능은 트래픽 증가에 대응할 수 있어 운영 오버헤드가 적습니다.

◆ | Q#0505. | Ref#0505.

한 회사에서 온프레미스 IoT 플랫폼을 AWS로 마이그레이션하고 있습니다. 이 플랫폼은 다음과 같은 구성 요소로 구성되어 있습니다.

- 수집 및 처리된 모든 IoT 데이터의 데이터 저장소인 MongoDB 클러스터.
- Message Queuing Telemetry Transport(MQTT)를 사용하여 5분마다 IoT 기기에 연결하여 데이터를 수집하는 애플리케이션.
- IoT 데이터에서 보고서를 생성하기 위해 주기적으로 작업을 실행하는 애플리케이션. 이 작업은 완료되는 데 120~600초가 걸립니다.
- 웹 서버에서 실행되는 웹 애플리케이션. 최종 사용자는 웹 애플리케이션을 사용하여 일반 대중이 액세스할 수 있는 보고서를 생성합니다.

이 회사는 성능을 유지하면서 운영 오버헤드를 줄이기 위해 플랫폼을 AWS로 마이그레이션해야 합니다.

이러한 요구 사항을 가장 적은 운영 오버헤드로 충족하는 단계 조합은 무엇입니까? (세 가지를 선택하십시오.)

- A.** AWS Lambda 작업으로 AWS Step Functions 상태 머신을 만들어 보고서를 준비하고 Amazon S3에 보고서를 작성합니다. 보고서를 제공하기 위해 S3 출처가 있는 Amazon CloudFront 배포를 구성합니다.
- B.** AWS Lambda 함수를 만듭니다. Lambda 함수를 프로그래밍하여 IoT 기기에 연결합니다. 데이터를 처리하고, 데이터를 데이터 저장소에 씁니다. 처리를 위해 메시지를 일시적으로 저장하도록 Lambda 계층을 구성합니다.
- C.** Amazon EC2 인스턴스로 Amazon Elastic Kubernetes Service(Amazon EKS) 클러스터를 구성하여 보고서를 준비합니다. EKS 클러스터에 인그레스 컨트롤러를 생성하여 보고서를 제공합니다.

- D.** IoT 기기를 AWS IoT Core에 연결하여 메시지를 게시합니다. 메시지를 수신하면 실행되는 AWS IoT 규칙을 만듭니다. AWS Lambda 함수를 호출하도록 규칙을 구성합니다. Lambda 함수를 프로그래밍하여 기기 메시지 데이터를 구문 분석, 변환하고 데이터 저장소에 저장합니다.
- E.** MongoDB 클러스터를 Amazon DocumentDB(MongoDB와 호환)로 마이그레이션합니다.
- F.** MongoDB 클러스터를 Amazon EC2 인스턴스로 마이그레이션합니다.

해설

정답: A,D,E

운영 오버헤드를 최소화하고 성능을 유지하려면 다음과 같은 조합이 적합합니다:

옵션 A: AWS Step Functions를 사용하여 보고서 생성 작업을 조정할 수 있으며, AWS Lambda를 사용하여 보고서를 생성하고 Amazon S3에 저장할 수 있습니다. Amazon CloudFront를 사용하여 S3에 저장된 보고서를 사용자에게 빠르게 제공할 수 있습니다. 이 조합은 운영 오버헤드를 줄이고, 보고서 생성과 배포를 자동화하며, 성능을 유지합니다.

옵션 D: AWS IoT Core를 사용하여 IoT 장치와 연결하고, AWS IoT 규칙을 설정하여 Lambda 함수를 호출하여 메시지를 처리하고 저장소에 데이터를 저장할 수 있습니다. 이 방법은 IoT 데이터의 수집과 처리를 자동화하고 운영 오버헤드를 줄일 수 있습니다.

옵션 E: Amazon DocumentDB는 MongoDB와 호환되는 관리형 데이터베이스 서비스입니다.

MongoDB를 DocumentDB로 마이그레이션하면 관리 오버헤드를 줄일 수 있으며, MongoDB 호환성도 유지됩니다. 이 옵션은 데이터 저장소에 대한 운영 오버헤드를 줄이면서 성능을 유지할 수 있습니다.

◆ | Q#0506. | Ref#0506.

한 회사가 Amazon API Gateway API를 만들고 외부 개발 팀과 API를 공유합니다. API는 AWS Lambda 함수를 사용하고 Production이라는 단계에 배포됩니다.

외부 개발 팀은 API의 유일한 소비자입니다. API는 특정 시간에 사용량이 갑자기 증가하여 비용 증가에 대한 우려가 발생합니다. 이 회사는 Lambda 함수를 다시 작업하지 않고도 비용과 사용량을 제한해야 합니다.

이러한 요구 사항을 가장 비용 효율적으로 충족하는 솔루션은 무엇입니까?

- A.** Lambda 함수에 직접 요청을 보내는 대신 Amazon Simple Queue Service(Amazon SQS) 대기열에 요청을 보내도록 API를 구성합니다. 대기열에서 메시지를 소비하고 요청을 처리하도록 Lambda 함수를 업데이트합니다. 새 메시지가 도착하면 Lambda 함수를 호출하도록 대기열을 설정합니다.
- B.** 각 Lambda 함수에 대해 프로비저닝된 동시성을 구성합니다. AWS Application Auto Scaling을 사용하여 Lambda 함수를 대상으로 등록합니다. API 사용의 변화에 맞춰 용량을 늘리거나 줄이도록 확장 일정을 설정합니다.
- C.** API Gateway API 키와 AWS WAF Regional 웹 ACL을 만듭니다. 웹 ACL을 Production 단계에 연결합니다. 웹 ACL에 요율 기반 규칙을 추가합니다. 규칙에서 요율 제한과 X-API-Key 헤더를 사용하는 사용자 지정 요청 집계를 지정합니다. API 키를 외부 개발 팀과 공유합니다.
- D.** API Gateway API 키와 사용 계획을 만듭니다. 사용 계획에서 제한 및 할당량을 정의합니다. 사용 계획을 프로덕션 단계 및 API 키와 연결합니다. API 키를 외부 개발 팀과 공유합니다.

해설

정답: D

가장 비용 효율적으로 API 사용량과 비용을 제한할 수 있는 방법은 옵션 D입니다. API Gateway의 Usage Plan을 사용하여 API 호출에 대한 스로틀링 및 쿼터를 설정함으로써 비용과 사용량을 효과적으로 관리할 수 있습니다. 이는 Lambda 함수와 API Gateway의 비용을 직접적으로 제어하며, 추가적인 복잡성 없이 요구 사항을 충족할 수 있습니다.

API Gateway의 Usage Plan을 사용하여 API 호출에 대한 스로틀링 및 쿼터를 설정할 수 있습니다. 스로틀링은 단위 시간당 허용된 요청 수를 제한하고, 쿼터는 특정 기간 동안의 총 요청 수를 제한합니다. 이를 통해 비용을 제어하고 API의 사용량을 제한할 수 있습니다.

이 방법은 비용을 직접적으로 제한하고 API 사용량을 제어할 수 있는 가장 효율적인 방법입니다.

◆ | Q#0507. | Ref#0507.

엔터테인먼트 회사가 Auto Scaling 그룹에 있는 Linux Amazon EC2 인스턴스의 함대에서 티켓팅 서비스를 호스팅합니다. 티켓팅 서비스는 가격 파일을 사용합니다. 가격 파일은 S3 Standard 스토리지가 있는 Amazon S3 버킷에 저장됩니다. 타사에서 호스팅하는 중앙 가격 솔루션이 가격 파일을 업데이트합니다.

가격 파일은 1~15분마다 업데이트되며 수천 개의 라인 항목이 있습니다. 가격 파일은 인스턴스가 시작될 때 각 EC2 인스턴스에 다운로드됩니다.

EC2 인스턴스는 때때로 오래된 가격 정보를 사용하여 고객에게 잘못된 요금이 청구될 수 있습니다.

어떤 솔루션이 이 문제를 가장 비용 효율적으로 해결할까요?

- A.** 가격 파일이 업데이트될 때마다 Amazon DynamoDB 테이블을 새 가격으로 업데이트하는 AWS Lambda 함수를 만듭니다. 티켓팅 서비스를 업데이트하여 DynramoDB를 사용하여 가격을 조회합니다.
- B.** 파일이 업데이트될 때마다 Amazon Elastic File System(Amazon EFS) 파일 공유를 가격 책정 파일로 업데이트하는 AWS Lambda 함수를 만듭니다. 티켓팅 서비스를 업데이트하여 Amazon EFS를 사용하여 가격 책정 파일에 액세스합니다.
- C.** EC2 인스턴스의 AMI에 Amazon S3용 Mountpoint를 로드합니다. Amazon S3용 Mountpoint를 구성하여 가격 파일이 포함된 S3 버킷을 마운트합니다. 티켓팅 서비스를 업데이트하여 마운트 지점과 S3 객체에 액세스하는 경로를 가리킵니다.
- D.** Amazon Elastic Block Store(Amazon EBS) 볼륨을 만듭니다. EBS Multi-Attach를 사용하여 모든 EC2 인스턴스에 볼륨을 연결합니다. 새 EC2 인스턴스가 시작되면 새 인스턴스를 구성하여 EBS 볼륨의 가격 파일을 업데이트합니다. 티켓팅 서비스를 업데이트하여 새 로컬 소스를 가리킵니다.

해설

정답: C

가격 파일이 자주 업데이트되고 최신 정보를 유지하는 것이 중요하기 때문에, 가장 비용 효율적이고 실시간 동기화를 지원하는 방법은 옵션 C입니다. Mountpoint for Amazon S3를 사용하여 S3 버킷을 EC2 인스턴스에 마운트하면, 인스턴스에서 최신 가격 파일을 직접 접근할 수 있으며, 파일을 개별적으로 다운로드할 필요가 없습니다. 이는 유지 관리가 용이하고, 비용 효율적이며, 파일 동기화 문제를 효과적으로 해결할 수 있습니다.

Mountpoint for Amazon S3를 사용하여 S3 버킷을 EC2 인스턴스에 마운트하면 인스턴스에서 S3 객체를 파일 시스템처럼 사용할 수 있습니다.

이는 S3에 저장된 최신 파일을 실시간으로 사용할 수 있게 하여, 파일을 개별적으로 다운로드할 필요가 없습니다.

이 방법은 비용이 적고, S3와의 통합을 쉽게 할 수 있습니다.

◆ | Q#0508. | Ref#0508.

한 회사에 Auto Scaling 그룹에서 Amazon EC2 인스턴스를 사용하는 애플리케이션이 있습니다. 품질 보증(QA) 부서는 애플리케이션을 테스트하기 위해 많은 수의 단기 환경을 시작해야 합니다. 애플리케이션 환경은 현재 부서 관리자가 AWS CloudFormation 템플릿을 사용하여 시작합니다. 스택을 시작하기 위해 관리자는 CloudFormation, EC2 및 Auto Scaling API를 사용할 수 있는 권한이 있는 역할을 사용합니다. 관리자는 테스터가 자신의 환경을 시작하도록 허용하고 싶지만 각 사용자에게 광범위한 권한을 부여하고 싶지 않습니다.

어떤 설정이 이러한 목표를 달성할까요?

- A.** AWS CloudFormation 템플릿을 Amazon S3에 업로드합니다. QA 부서의 사용자에게 관리자 역할을 맡을 수 있는 권한을 부여하고 템플릿과 템플릿이 생성하는 리소스에 대한 권한을 제한하는 정책을 추가합니다. 사용자에게 CloudFormation 콘솔에서 템플릿을 시작하도록 교육합니다.
- B.** 환경 템플릿에서 AWS Service Catalog 제품을 만듭니다. 기존 역할로 제품에 시작 제약 조건을 추가합니다. QA 부서의 사용자에게 AWS Service Catalog API만 사용할 수 있는 권한을 부여합니다. 사용자에게 AWS Service Catalog 콘솔에서 템플릿을 시작하도록 교육합니다.
- C.** AWS CloudFormation 템플릿을 Amazon S3에 업로드합니다. QA 부서의 사용자에게

CloudFormation 및 S3 API를 사용할 수 있는 권한을 부여하고, 템플릿과 템플릿이 생성하는 리소스에 대한 권한을 제한하는 조건을 적용합니다. 사용자에게 CloudFormation 콘솔에서 템플릿을 시작하도록 교육합니다.

D. 환경 템플릿에서 AWS Elastic Beanstalk 애플리케이션을 만듭니다. QA 부서의 사용자에게 Elastic Beanstalk 권한만 사용할 수 있는 권한을 부여합니다. 사용자에게 Elastic Beanstalk CLI로 Elastic Beanstalk 환경을 시작하도록 교육하고 기존 역할을 서비스 역할로 환경에 전달합니다.

해설

정답: B

가장 적절한 방법은 옵션 B입니다. AWS Service Catalog를 사용하여 템플릿을 제품으로 만들고, 서비스 카탈로그 API만을 사용하여 제한된 권한을 부여함으로써 관리자가 템플릿과 환경을 제어할 수 있습니다. 이 방법은 사용자에게 필요한 권한만 부여하며, 보안성과 관리의 편리함을 높일 수 있습니다.

AWS Service Catalog를 사용하여 템플릿을 제품으로 변환하면, 제품 카탈로그에서 정의된 권한과 제약을 관리할 수 있습니다.

서비스 카탈로그는 사용자에게 CloudFormation 템플릿을 실행할 수 있는 권한을 부여하되, IAM 역할과 권한을 제어할 수 있도록 지원합니다. 이 방법은 테스터들에게 필요한 권한만 부여하고 관리할 수 있어 보안성과 관리 편의성을 높일 수 있습니다.

사용자들에게 서비스 카탈로그 콘솔을 통해 템플릿을 실행하도록 교육하면 됩니다.

◆ | Q#0509. | Ref#0509.

한 회사가 전자상거래 웹사이트에 단일 AWS 리전을 사용하고 있습니다. 이 웹사이트에는 Application Load Balancer(ALB) 뒤의 여러 Amazon EC2 인스턴스에서 실행되는 웹 애플리케이션이 포함되어 있습니다. 이 웹사이트에는 Amazon DynamoDB 테이블도 포함되어 있습니다. Amazon Route 53의 사용자 지정 도메인 이름이 ALB에 연결되어 있습니다. 이 회사는 AWS Certificate Manager(ACM)에서 SSL/TLS 인증서를 생성하여 ALB에 첨부했습니다. 이 회사는 설계의 일부로 콘텐츠 전송 네트워크를 사용하지 않습니다.

이 회사는 재해 복구를 제공하고, 향후 성장을 계획하고, 사용자에게 향상된 액세스 시간을 제공하기 위해 두 번째 리전에 전체 애플리케이션 스택을 복제하려고 합니다. 솔루션 아키텍트는 이러한 목표를 달성하고 관리 오버헤드를 최소화하는 솔루션을 구현해야 합니다. 솔루션 아키텍트는 이러한 요구 사항을 충족하기

위해 어떤 단계 조합을 취해야 할까요? (세 가지를 선택하십시오.)

- A.** 현재 인프라 설계에 대한 AWS CloudFormation 템플릿을 만듭니다. Region을 포함한 중요한 시스템 값에 대한 매개변수를 사용합니다. CloudFormation 템플릿을 사용하여 두 번째 Region에 새 인프라를 만듭니다.
- B.** AWS Management Console을 사용하여 첫 번째 지역의 기존 인프라 설계를 문서화하고 두 번째 지역에 새 인프라를 생성합니다.
- C.** 애플리케이션이 가중치 라우팅을 사용하도록 Route 53 호스팅 영역 레코드를 업데이트합니다. 각 지역의 ALB로 트래픽의 50%를 보냅니다.
- D.** 애플리케이션이 대기 시간 기반 라우팅을 사용하도록 Route 53 호스팅 영역 레코드를 업데이트합니다. 각 지역의 ALB로 트래픽을 보냅니다.
- E.** DynamoDB Streams를 활성화하여 기존 DynamoDB 테이블의 구성을 업데이트합니다. 두 번째 Region을 추가하여 글로벌 테이블을 만듭니다.
- F.** 새 DynamoDB 테이블을 만듭니다. 새 테이블에 DynamoDB Streams를 활성화합니다. 두 번째 Region을 추가하여 글로벌 테이블을 만듭니다. 기존 DynamoDB 테이블의 데이터를 일회성 작업으로 새 테이블로 복사합니다.

해설

정답: A,D,E

옵션 A: AWS CloudFormation을 사용하여 현재 인프라를 템플릿으로 만들고, 이를 두 번째 리전에서 재사용하면, 일관성 있는 환경을 쉽게 배포할 수 있습니다. 파라미터를 사용하여 지역과 같은 중요한 시스템 값을 설정하면 템플릿을 유연하게 사용할 수 있습니다. 이 방법은 효율적이고 관리 오버

헤드를 최소화하는 데 유용합니다.

옵션 D: Latency-based routing은 사용자의 지리적 위치에 따라 가장 빠른 응답 시간을 제공하는 리전으로 트래픽을 자동으로 전송합니다. 이는 성능 개선에 유리하며, 트래픽을 효율적으로 분산시킬 수 있습니다. 이 방법은 사용자 접근 속도를 개선하고, 두 리전 간의 로드 밸런싱을 효과적으로 수행할 수 있습니다.

옵션 E: DynamoDB 글로벌 테이블을 사용하면 여러 리전에 걸쳐 데이터를 자동으로 동기화할 수 있습니다. 이는 데이터 복제와고가용성을 제공하지만, 이미 존재하는 테이블에 대해 설정을 업데이트하는 방법입니다.

◆ | Q#0510. | Ref#0510.

한 회사에서 데이터 과학자가 업무 관련 문서를 저장할 단일 Amazon S3 버킷을 만들고자 합니다. 이 회사는 AWS IAM Identity Center를 사용하여 모든 사용자를 인증합니다. 데이터 과학자를 위한 그룹이 생성되었습니다.

이 회사는 데이터 과학자에게 자신의 작업에만 액세스할 수 있도록 하려고 합니다. 또한 이 회사는 각 사용자가 액세스한 문서를 보여주는 월별 보고서를 만들고자 합니다.

이러한 요구 사항을 충족하는 단계 조합은 무엇입니까? (두 가지를 선택하세요.)

- A.** 사용자 정의 IAM Identity Center 권한 세트를 만들어 데이터 과학자에게 사용자 이름 태그와 일치하는 S3 버킷 접두사에 대한 액세스 권한을 부여합니다. 정책을 사용하여 `${aws:PrincipalTag/userName}/*` 조건이 있는 경로에 대한 액세스를 제한합니다.
- B.** Amazon S3 읽기 액세스 및 쓰기 액세스가 있는 데이터 과학자 그룹에 대한 IAM Identity Center 역할을 만듭니다. IAM Identity Center 역할에 대한 액세스를 허용하는 S3 버킷 정책을 추가합니다.
- C.** AWS CloudTrail을 구성하여 S3 데이터 이벤트를 로깅하고 로그를 S3 버킷에 전달합니다. Amazon Athena를 사용하여 Amazon S3의 CloudTrail 로그에 대한 쿼리를 실행하고 보고서를 생성합니다.
- D.** AWS CloudTrail을 구성하여 S3 관리 이벤트를 CloudWatch에 기록합니다. Amazon Athena의 CloudWatch 커넥터를 사용하여 로그를 쿼리하고 보고서를 생성합니다.
- E.** EMR 파일 시스템(EMRFS)에 대한 S3 액세스 로깅을 활성화합니다. Amazon S3 Select를 사용하여 로그를 쿼리하고 보고서를 생성합니다.

해설

정답: A,C

옵션 A: IAM Identity Center에서 사용자 태그를 기반으로 S3 버킷의 접근을 제어할 수 있습니다. 이 옵션은 각 데이터 과학자에게 자신의 사용자 이름 태그를 포함하는 버킷 경로에만 접근할 수 있도록 하는 방법을 제시합니다. 정책에서 `${aws:PrincipalTag/userName}/*` 조건을 사용하면 사용자가 자신의 작업만 접근할 수 있도록 할 수 있습니다. 이는 각 사용자의 데이터 접근을 적절히 제한할 수 있습니다.

옵션 C: AWS CloudTrail을 사용하여 S3 데이터 이벤트를 기록하고, 이를 S3 버킷에 전달하여 Amazon Athena로 쿼리하여 보고서를 생성할 수 있습니다. 이는 각 사용자가 접근한 문서에 대한 기록을 수집하고 분석하는 데 유용합니다. CloudTrail 로그를 분석하면 사용자의 접근 패턴을 추적할 수 있습니다.

511 (노종옥,최정현) 5회차 完

◆ | Q#0511. | Ref#0511.

한 회사가 Amazon EC2 인스턴스에서 데이터 처리 애플리케이션을 호스팅합니다. 이 애플리케이션은 새로 업로드된 파일에 대해 Amazon Elastic File System(Amazon EFS) 파일 시스템을 폴링합니다. 새 파일이 감지되면 애플리케이션은 파일에서 데이터를 추출하고 파일을 처리할 Docker 컨테이너 이미지를 선택하는 로직을 실행합니다. 애플리케이션은 적절한 컨테이너 이미지를 시작하고 파일 위치를 매개변수로 전달합니다.

컨테이너가 수행하는 데이터 처리에는 최대 2시간이 걸릴 수 있습니다. 처리가 완료되면 컨테이너 내부에서 실행되는 코드가 파일을 Amazon EFS에 다시 쓰고 종료됩니다.

이 회사는 컨테이너를 실행하는 EC2 인스턴스를 제거하기 위해 애플리케이션을 리팩터링해야 합니다.

이러한 요구 사항을 충족하는 솔루션은 무엇입니까?

- A.** Amazon Elastic Container Service(Amazon ECS) 클러스터를 만듭니다. AWS Fargate 작업으로 실행되도록 처리를 구성합니다. 컨테이너 선택 로직을 추출하여 적절한 Fargate 작업을 시작하는 Amazon EventBridge 규칙으로 실행합니다. 파일이 EFS 파일 시스템에 추가될 때 실행되도록 EventBridge 규칙을 구성합니다.
- B.** Amazon Elastic Container Service(Amazon ECS) 클러스터를 만듭니다. AWS Fargate 작업으로 실행되도록 처리를 구성합니다. 컨테이너 선택 로직을 업데이트하고 컨테이너화하여 적절한 Fargate 작업을 시작하는 Fargate 서비스로 실행합니다. EFS 파일 시스템에 파일이 추가되면 Fargate 서비스를 호출하도록 EFS 이벤트 알림을 구성합니다.
- C.** Amazon Elastic Container Service(Amazon ECS) 클러스터를 만듭니다. AWS Fargate 작업으로 실행되도록 처리를 구성합니다. 컨테이너 선택 로직을 추출하여 적절한 Fargate 작업을 시작하는 AWS Lambda 함수로 실행합니다. 파일 업로드의 저장소를 Amazon S3 버킷으로 마이그레이션합니다. Amazon S3를 사용하도록 처리 코드를 업데이트합니다. 객체가 생성될 때 Lambda 함수를 호출하도록 S3 이벤트 알림을 구성합니다.
- D.** 처리를 위한 AWS Lambda 컨테이너 이미지를 만듭니다. 컨테이너 이미지를 사용하도록 Lambda 함수를 구성합니다. 컨테이너 선택 로직을 추출하여 적절한 Lambda 처리 함수를 호출하는 결정 Lambda 함수로 실행합니다. 파일 업로드의 스토리지를 Amazon S3 버킷으로 마이그레이션합니다. Amazon S3를 사용하도록 처리 코드를 업데이트합니다. 객체가 생성될 때 결정 Lambda 함수를 호출하도록 S3 이벤트 알림을 구성합니다.

해설

정답: C

AWS Fargate 구성 > AWS Lambda 함수로 실행 > 파일 업로드의 저장소를 Amazon S3 버킷 > Lambda 함수를 호출하도록 S3 이벤트 알림을 구성

◆ | Q#0512. | Ref#0512.

미디어 회사에 디지털 뉴스 비디오의 30-T8 저장소가 있습니다. 이러한 비디오는 온프레미스 테이프 라이브러리의 테이프에 저장되고 미디어 자산 관리(MAM) 시스템에서 참조됩니다. 이 회사는 MAM 기능을 사용하여 이러한 비디오의 메타데이터를 자동화된 방식으로 풍부하게 하고 검색 가능한 카탈로그에 넣으려고 합니다. 이 회사는 객체, 풍경 항목 또는 사람의 얼굴과 같은 비디오의 정보를 기반으로 검색할 수 있어야 합니다. 각 사람의 이미지가 포함된 비디오에 등장한 사람의 얼굴이 들어 있는 카탈로그를 사용할 수 있습니다. 이 회사는 이러한 비디오를 AWS로 마이그레이션하려고 합니다.

이 회사는 AWS와 고속 AWS Direct Connect 연결을 사용하고 있으며 MAM 솔루션 비디오 콘텐츠를 현재 파일 시스템에서 직접 옮기려고 합니다.

지속적인 관리 오버헤드를 최소화하고 기존 시스템에 최소한의 중단을 발생시켜 이러한 요구 사항을 어떻게 충족할 수 있습니까?

- A.** 온프레미스에서 AWS Storage Gateway, 파일 게이트웨이 어플라이언스를 설정합니다. MAM 솔루션을 사용하여 현재 아카이브에서 비디오를 추출하여 파일 게이트웨이로 푸시합니다. 얼굴 카탈로그를 사용하여 Amazon Rekognition에서 컬렉션을 빌드합니다. Rekognition Javascript SDK를 호출하여 Rekognition이 파일 게이트웨이를 백업하는 Amazon S3 파일에서 비디오를 가져오고, 필요한 메타데이터를 검색하여 메타데이터를 MAM 솔루션으로 푸시하도록 하는 AWS Lambda 함수를 빌드합니다.
- B.** 온프레미스에서 AWS Storage Gateway, 테이프 게이트웨이 어플라이언스를 설정합니다. MAM 솔루션을 사용하여 현재 아카이브에서 비디오를 추출하여 테이프 게이트웨이로 푸시합니다. 얼굴 카탈로그를 사용하여 Amazon Rekognition에서 컬렉션을 빌드합니다. Rekognition Javascript SDK를 호출하여 Amazon Rekognition이 테이프 게이트웨이에서 비디오를 처리하고 필요한 메타데이터를 검색하여 메타데이터를 MAM 솔루션으로 푸시하도록 하는 AWS Lambda 함수를 빌드합니다.

C. Amazon Kinesis Video Streams를 사용하여 비디오 수집 스트림을 구성합니다. 얼굴 카탈로그를 사용하여 Amazon Rekognition에서 컬렉션을 빌드합니다. MAM 솔루션에서 Kinesis Video Streams로 비디오를 스트리밍합니다. Amazon Rekognition을 구성하여 스트리밍된 비디오를 처리합니다. 그런 다음 스트림 소비자를 사용하여 필요한 메타데이터를 검색하고 메타데이터를 MAM 솔루션으로 푸시합니다. 스트림을 구성하여 Amazon S3에 비디오를 저장합니다.

D. OpenCV 라이브러리를 실행하는 Amazon EC2 인스턴스를 설정합니다. 온프레미스 라이브러리에서 비디오, 이미지 및 얼굴 카탈로그를 이 EC2 인스턴스에 마운트된 Amazon EBS 볼륨으로 복사합니다. 비디오를 처리하여 필요한 메타데이터를 검색하고 메타데이터를 MAM 솔루션으로 푸시하는 동시에 비디오 파일을 Amazon S3 버킷으로 복사합니다.

해설

정답:B

온프레미스 테이프 라이브러리의 테이프에 저장 -> 온프레미스에서 AWS Storage Gateway, 테이프 게이트웨이 어플라이언스를 설정

◆ | Q#0513. | Ref#0513.

한 회사는 AWS Organizations에서 여러 계정이 포함된 AWS 환경의 비용을 최적화해야 합니다. 이 회사는 3년 전에 비용 최적화 활동을 수행했고 최근 만료된 Amazon EC2 Standard Reserved Instances를 구매했습니다.

이 회사는 3년 더 EC2 인스턴스가 필요합니다. 또한 이 회사는 새로운 서버리스 워크로드를 배포했습니다.

어떤 전략이 회사에 가장 많은 비용 절감을 제공할까요?

- A.** 모든 선불 결제로 3년 추가 기간 동안 동일한 예약 인스턴스를 구매합니다. 모든 선불 결제로 관리 계정에서 3년 Compute Savings Plan을 구매하여 추가 컴퓨팅 비용을 충당합니다.
- B.** 각 멤버 계정에서 선불 없이 1년 Compute Savings Plan을 구매합니다. AWS Cost Management 콘솔에서 Savings Plans 권장 사항을 사용하여 Compute Savings Plan을 선택합니다.
- C.** 각 AWS 지역의 EC2 비용을 충당하기 위해 관리 계정에서 선불 없이 3년 EC2 인스턴스 절약 플랜을 구매합니다. 추가 컴퓨팅 비용을 충당하기 위해 관리 계정에서 선불 없이 3년 컴퓨팅 절약 플랜을 구매합니다.
- D.** 각 멤버 계정에서 모든 선불 결제로 3년 EC2 인스턴스 절약 플랜을 구매합니다. AWS Cost Management 콘솔에서 절약 플랜 권장 사항을 사용하여 EC2 인스턴스 절약 플랜을 선택합니다.

해설

정답:A

가장 많은 비용 절감 -> 불금이 높을수록 할인도 커집니다. EC2 인스턴스를 실행하는 가장 저렴한 방법은 예약 인스턴스 또는 EC2 인스턴스 절약 플랜

◆ | Q#0514. | Ref#0514.

한 회사가 전 세계 고객에게 서비스를 제공하는 정적 콘텐츠 배포 플랫폼을 운영합니다. 고객은 자체 AWS 계정에서 콘텐츠를 소비합니다.

이 회사는 Amazon S3 버킷에서 콘텐츠를 제공합니다. 이 회사는 S3 파일 게이트웨이를 사용하여 온프레미스 환경에서 S3 버킷으로 콘텐츠를 업로드합니다.

이 회사는 지리적으로 고객과 가장 가까운 AWS 리전에서 콘텐츠를 제공하여 플랫폼의 성능과 안정성을 개선하고자 합니다. 이 회사는 최소한의 지연 시간과 공개 인터넷 노출 없이 온프레미스 데이터를 Amazon S3로 라우팅해야 합니다.

어떤 단계 조합이 최소한의 운영 오버헤드로 이러한 요구 사항을 충족할 수 있을까요? (두 가지를 선택하세요.)

- A.** S3 다중 지역 액세스 포인트 구현
- B.** S3 Cross-Region Replication(CRR)을 사용하여 콘텐츠를 다른 지역으로 복사합니다.
- C.** 클라이언트의 지역 라우팅을 추적하는 AWS Lambda 함수를 만듭니다.

D. AWS 사이트 간 VPN 연결을 사용하여 다중 지역 액세스 포인트에 연결합니다.

E. AWS PrivateLink와 AWS Direct Connect를 사용하여 다중 지역 액세스 포인트에 연결합니다.

해설

정답:AE

최소한의 지연 시간+공개인터넷노출없이 -> S3 다중 지역 액세스 포인트 구현, AWS PrivateLink와 AWS Direct Connect를 사용

◆ | Q#0515. | Ref#0515.

한 회사가 데이터 센터를 AWS 클라우드로 마이그레이션하고 있으며 가능한 한 빨리 마이그레이션을 완료해야 합니다. 이 회사에는 데이터 센터에 있는 수백 개의 VMware VM에서 실행되는 많은 애플리케이션이 있습니다. 각 VM은 공통 공유 파일이 포함된 공유 Windows 폴더로 구성되어 있습니다. 파일 공유는 크기가 100GB보다 큼니다.

이 회사의 규정 준수 팀은 각 VM에 대한 모든 소프트웨어 설치 및 수정에 대해 변경 요청을 제출하고 승인해야 합니다. 이 회사는 AWS와 데이터 센터 간에 10GB 대역폭의 AWS Direct Connect 연결을 보유하고 있습니다.

이 회사는 최소한의 시간 내에 마이그레이션을 완료하기 위해 어떤 단계를 거쳐야 합니까?

A. VM ImportExport를 사용하여 각 VM의 이미지를 만듭니다. AWS Application Migration Service를 사용하여 이미지를 관리하고 복제합니다. Windows 파일 공유 데이터를 Amazon Elastic File System(Amazon EFS) 파일 시스템에 복사합니다. 마이그레이션 후 파일 공유를 EFS 파일 시스템에 다시 매핑합니다.

B. AWS Application Discovery Service 에이전트리스 어플라이언스를 VMware vCenter에 배포합니다. AWS Migration Hub에서 검색된 VM의 포트폴리오를 검토합니다.

C. AWS Application Migration Service 에이전트리스 어플라이언스를 VMware vCenter에 배포합니다. Windows 파일 공유 데이터를 새로운 Amazon FSx for Windows File Server 파일 시스템에 복사합니다. 마이그레이션 후 각 VM의 파일 공유를 FSx for Windows File Server 파일 시스템에 다시 매핑합니다.

D. AWS Migration Hub에서 포트폴리오를 만들고 검토합니다. AWS Snowcone 디바이스를 주문합니다. AWS Application Migration Service를 VMware vCenter에 배포하고 모든 VM을 Snowcone 디바이스로 내보냅니다. 모든 Windows 파일 공유 데이터를 Snowcone 디바이스에 복사합니다. Snowcone 디바이스를 AWS로 배송합니다. Application Migration Service를 사용하여 모든 마이그레이션된 인스턴스를 배포합니다.

E. AWS Application Discovery Service Agent와 AWS Application Migration Service Agent를 각 VMware 하이퍼바이저에 직접 배포합니다. AWS Migration Hub에서 포트폴리오를 검토합니다. 각 VM의 파일 공유 데이터를 새 Amazon FSx for Windows File Server 파일 시스템에 복사합니다. 마이그레이션 후 각 VM의 파일 공유를 FSx for Windows File Server 파일 시스템에 다시 매핑합니다.

해설

정답:C

최소한의 시간 내에 마이그레이션을 완료 -> AWS Application Migration Service 에이전트리스 어플라이언스를 VMware vCenter에 배포
공유 Windows 폴더로 구성 -> Amazon FSx for Windows File Server

◆ | Q#0516. | Ref#0516.

한 회사에는 AWS Organizations의 조직에 있는 여러 AWS 계정이 있습니다. 이 회사는 AWS 계정 활동을 저장하고 SQL을 사용하여 중앙 위치에서 데이터를 쿼리해야 합니다.

어떤 솔루션이 이러한 요구 사항을 충족할까요?

A. 각 계정에서 AWS CloudTrail 트레일을 만듭니다. 트레일에 대한 CloudTrail 관리 이벤트를 지정합니다. 이벤트를 Amazon CloudWatch Logs로 보내도록 CloudTrail을 구성합니다. CloudWatch 교차 계정 관찰을 구성합니다. CloudWatch Logs Insights에서 데이터를 쿼리합니다.

B. 위임된 관리자 계정을 사용하여 AWS CloudTrail Lake 데이터 스토어를 만듭니다. 데이터 스토어

에 대한 CloudTrail 관리 이벤트를 지정합니다. 조직의 모든 계정에 대한 데이터 스토어를 활성화합니다. CloudTrail Lake에서 데이터를 쿼리합니다.

C. 위임된 관리자 계정을 사용하여 AWS CloudTrail 트레일을 만듭니다. 트레일에 대한 CloudTrail 관리 이벤트를 지정합니다. 조직의 모든 계정에 대한 트레일을 활성화합니다. 다른 모든 설정은 기본값으로 유지합니다. CloudTrail 이벤트 기록 페이지에서 CloudTrail 데이터를 쿼리합니다.

D. AWS CloudFormation StackSets를 사용하여 각 계정에서 AWS CloudTrail Lake 데이터 스토어를 배포합니다. 데이터 스토어에 대한 CloudTrail 관리 이벤트를 지정합니다. 다른 모든 설정은 기본값으로 유지하고 CloudTrail Lake에서 데이터를 쿼리합니다.

해설

정답:B

AWS CloudTrail Lake를 사용하여 조직의 모든 계정에 대해 이벤트를 집계하고, 이벤트에 대한 SQL 기반 쿼리를 실행할 수 있습니다.

◆ | Q#0517. | Ref#0517.

한 회사가 AWS를 사용하여 프로덕션 웹 애플리케이션을 개발하고 관리하고 있습니다. 이 애플리케이션에는 AWS Lambda 함수를 호출하는 Amazon API Gateway HTTP API가 포함되어 있습니다. Lambda 함수는 데이터를 처리한 다음 데이터베이스에 저장합니다.

이 회사는 웹 애플리케이션에 대한 사용자 권한을 통합된 방식으로 구현하려고 합니다. 이 회사는 이미 회사의 다른 애플리케이션에 대한 OAuth 토큰을 발급하는 타사 ID 공급자를 사용하고 있습니다.

어떤 솔루션이 이러한 요구 사항을 충족할까요?

A. 회사의 타사 ID 공급자를 API Gateway와 통합합니다. ID 공급자의 토큰을 검증하도록 API Gateway Lambda 권한 부여자를 구성합니다. 모든 API 경로에서 Lambda 권한 부여자를 요구합니다. 웹 애플리케이션을 업데이트하여 ID 공급자로부터 토큰을 가져오고 API Gateway HTTP API를 호출할 때 Authorization 헤더에 토큰을 포함합니다.

B. 회사의 타사 ID 공급자를 AWS Directory Service와 통합합니다. Directory Service를 ID 공급자의 토큰을 검증하기 위한 API Gateway 권한 부여자로 구성합니다. 모든 API 경로에서 Directory Service 권한 부여자를 요구합니다. AWS IAM Identity Center를 SAML 2.0 ID 공급자로 구성합니다. 웹 애플리케이션을 사용자 지정 SAML 2.0 애플리케이션으로 구성합니다.

C. 회사의 타사 ID 공급자를 AWS IAM Identity Center와 통합합니다. API Gateway를 구성하여 IAM Identity Center를 사용하여 구성 없는 인증 및 권한 부여를 수행합니다. 웹 애플리케이션을 업데이트하여 IAM Identity Center에서 AWS Security Token Service(AWS STS) 토큰을 검색하고 API Gateway HTTP API를 호출할 때 Authorization 헤더에 토큰을 포함합니다.

D. 회사의 타사 ID 공급자를 AWS IAM Identity Center와 통합합니다. API Gateway HTTP API를 호출할 수 있는 권한이 있는 IAM 사용자를 구성합니다. 웹 애플리케이션을 업데이트하여 IAM 사용자로 부터 요청 매개변수를 추출하고 API Gateway HTTP API를 호출할 때 Authorization 헤더에 매개변수를 포함합니다.

해설

정답:A

API GW 통합 + Lambda Authorizer.

◆ | Q#0518. | Ref#0518.

한 회사가 AWS 계정의 수천 개의 Amazon EC2 인스턴스에 애플리케이션을 배포했습니다. 보안 감사에서 암호화되지 않은 여러 Amazon Elastic Block Store(Amazon EBS) 볼륨이 EC2 인스턴스에 연결되어 있음을 발견했습니다. 회사의 보안 정책에 따라 EBS 볼륨을 암호화해야 합니다.

회사는 EBS 볼륨을 암호화하는 자동화된 솔루션을 구현해야 합니다. 또한 솔루션은 개발팀이 암호화되지 않은 EBS 볼륨을 생성하지 못하도록 해야 합니다.

어떤 솔루션이 이러한 요구 사항을 충족할까요?

A. 암호화되지 않은 EBS 볼륨을 식별하는 AWS Config 관리 규칙을 구성합니다. 자동 수정 작업을 구성합니다. 새 암호화된 EBS 볼륨을 만드는 단계가 포함된 AWS Systems Manager Automation 런북을 연결합니다. AWS Key Management Service(AWS KMS) 고객 관리 키를 만듭니다. 키 정책에 암호화되지 않은 EBS 볼륨 생성을 거부하는 문장을 포함합니다.

B. AWS Systems Manager Fleet Manager를 사용하여 암호화되지 않은 EBS 볼륨 목록을 만듭니다. 암호화된 새 EBS 볼륨을 만드는 단계가 포함된 Systems Manager Automation 런북을 만듭니다. 암호화되지 않은 EBS 볼륨의 생성을 거부하는 SCP를 만듭니다.

C. AWS Systems Manager Fleet Manager를 사용하여 암호화되지 않은 EBS 볼륨 목록을 만듭니다. 새 암호화된 EBS 볼륨을 만드는 단계가 포함된 Systems Manager Automation 런북을 만듭니다. EBS 암호화에 대한 AWS 계정 설정을 수정하여 항상 새 EBS 볼륨을 암호화합니다.

D. 암호화되지 않은 EBS 볼륨을 식별하는 AWS Config 관리 규칙을 구성합니다. 자동 수정 작업을 구성합니다. 새 암호화된 EBS 볼륨을 만드는 단계가 포함된 AWS Systems Manager Automation 런북을 연결합니다. EBS 암호화에 대한 AWS 계정 설정을 수정하여 항상 새 EBS 볼륨을 암호화합니다.

해설

정답:D

AWS config를 사용하여 암호화되지 않은 EBS를 찾는다. > 자동수정작업을 구성, EBS에 대한 기본 암호화를 활성화

◆ | Q#0519. | Ref#0519.

한 회사가 AWS 클라우드에서 대규모 컨테이너화된 워크로드를 실행하고 있습니다. 워크로드는 약 100개의 서로 다른 서비스로 구성되어 있습니다. 이 회사는 Amazon Elastic Container Service(Amazon ECS)를 사용하여 워크로드를 조정합니다.

최근 이 회사의 개발팀은 ECS 클러스터에서 Amazon EC2 인스턴스 대신 AWS Fargate를 사용하기 시작했습니다. 과거에 워크로드는 계정에서 사용할 수 있는 최대 EC2 인스턴스 수에 가까워졌습니다.

이 회사는 워크로드가 허용되는 최대 ECS 작업 수에 도달할 수 있다는 점을 우려하고 있습니다. 솔루션 아키텍트는 Fargate가 최대 작업 수의 80%에 도달하면 개발팀에 알리는 솔루션을 구현해야 합니다.

솔루션 아키텍트는 이 요구 사항을 충족하기 위해 무엇을 해야 할까요?

A. Amazon CloudWatch를 사용하여 ECS 클러스터의 각 서비스에 대한 샘플 수 통계를 모니터링합니다. 수학 표현식 `샘플 수 / SERVICE_QUOTA(서비스)100이 80보다 클 때 알람을 설정합니다.`

Amazon Simple Notification Service(Amazon SNS)를 사용하여 개발 팀에 알립니다.

B. Amazon CloudWatch를 사용하여 AWS/Usage 메트릭 네임스페이스에 게시된 서비스 할당량을 모니터링합니다. 수학 표현식 `metric / SERVICE_QUOTA(metric)100이 80보다 클 때 알람을 설정합니다.` Amazon Simple Notification Service(Amazon SNS)를 사용하여 개발 팀에 알립니다.

C. ECS 클러스터에서 자세한 메트릭을 폴링하는 AWS Lambda 함수를 만듭니다. 실행 중인 Fargate 작업의 수가 80보다 큰 경우 Amazon Simple Email Service(Amazon SES)를 호출하여 개발 팀에 알립니다.

D. Fargate SERVICE_QUOTA가 80보다 큰지 평가하기 위한 AWS Config 규칙을 생성합니다. AWS Config 규칙이 규정을 준수하지 않는 경우 Amazon Simple Email Service(Amazon SES)를 사용하여 개발팀에 알립니다.

해설

정답:B

Amazon CloudWatch를 사용, 서비스 할당량을 모니터링, Amazon SNS를 사용하여 개발 팀에 알림

◆ | Q#0520. | Ref#0520.

한 회사에 Python으로 작성된 여러 AWS Lambda 함수가 있습니다. 이 함수는 .zip 패키지 배포 유형으로 배포됩니다. 이 함수는 .zip 파일에 공통 라이브러리와 패키지가 들어 있는 Lambda 계층을 사용합니다. Lambda .zip 패키지와 Lambda 계층 .zip 파일은 Amazon S3 버킷에 저장됩니다.

이 회사는 CVE를 식별하기 위해 Lambda 함수와 Lambda 계층의 자동 검사를 구현해야 합니다. Lambda 함수의 하위 집합은 잠재적인 데이터 유출 및 기타 취약성을 감지하기 위해 자동화된 코드 검사를 받아야 합니다. 코드 검사는 모든 Lambda 함수가 아닌 선택된 Lambda 함수에 대해서만 수행해야 합니다.

이러한 요구 사항을 충족하는 작업의 조합은 무엇입니까? (세 가지를 선택하십시오.)

- A.** Amazon Inspector를 활성화합니다. 자동화된 CVE 스캔을 시작합니다.
- B.** Amazon Inspector에서 Lambda 표준 스캐닝과 Lambda 코드 스캐닝을 활성화합니다.
- C.** Amazon GuardDuty 활성화. GuardDuty에서 Lambda Protection 기능 활성화.
- D.** 코드 스캔이 필요한 Lambda 함수의 모니터 설정에서 스캔을 활성화합니다.
- E.** 코드 스캔이 필요 없는 Lambda 함수에 태그를 지정합니다. 태그에 InspectorCodeExclusion 키와 LambdaCodeScanning 값을 포함합니다.
- F.** Amazon Inspector를 사용하여 Lambda .zip 패키지와 Lambda 계층 .zip 파일이 포함된 3개 버킷을 스캔하여 코드를 스캔합니다.

해설

정답:ABE

Amazon Inspector를 활성화, Lambda 표준 스캐닝과 Lambda 코드 스캐닝을 활성화, Lambda 함수에 태그를 지정

521 (나권서,박지수) 5회차 完

◆ | Q#0521. | Ref#0521.

한 회사가 애플리케이션 계정에서 Amazon EC2 인스턴스의 패치를 처리하는 방식을 변경하고 있습니다. 이 회사는 현재 애플리케이션 계정의 VPC에 있는 NAT 게이트웨이를 사용하여 인터넷을 통해 인스턴스에 패치를 적용하고 있습니다.

이 회사는 코어 계정의 전용 프라이빗 VPC에 EC2 인스턴스를 패치 소스 리포지토리로 설정했습니다. 이 회사는 AWS Systems Manager Patch Manager와 코어 계정의 패치 소스 리포지토리를 사용하여 애플리케이션 계정의 EC2 인스턴스에 패치를 적용하려고 합니다. 이 회사는 애플리케이션 계정의 모든 EC2 인스턴스가 인터넷에 액세스하지 못하도록 해야 합니다. 애플리케이션 계정의

EC2 인스턴스는 애플리케이션 데이터가 저장된 Amazon S3에 액세스해야 합니다. 이러한 EC2 인스턴스는 Systems Manager와 코어 계정의 프라이빗 VPC에 있는 패치 소스 리포지토리에 연결해야 합니다.

이러한 요구 사항을 충족하는 솔루션은 무엇입니까?

- A.** 포트 80에서 아웃바운드 트래픽을 차단하는 네트워크 ACL을 만듭니다. 네트워크 ACL을 애플리케이션 계정의 모든 서브넷과 연결합니다. 애플리케이션 계정과 코어 계정에서 사용자 지정 VPN 서버를 실행하는 EC2 인스턴스 하나를 배포합니다. 프라이빗 VPC에 액세스하기 위한 VPN 터널을 만듭니다. 애플리케이션 계정에서 경로 테이블을 업데이트합니다.
- B.** Systems Manager 및 Amazon S3에 대한 개인 VIF를 만듭니다. 애플리케이션 계정의 VPC에서 NAT 게이트웨이를 삭제합니다. 코어 계정의 패치 소스 리포지토리 EC2 인스턴스에 액세스하기 위한 전송 게이트웨이를 만듭니다. 코어 계정의 경로 테이블을 업데이트합니다.
- C.** Systems Manager와 Amazon S3에 대한 VPC 엔드포인트를 만듭니다. 애플리케이션 계정의 VPC에서 NAT 게이트웨이를 삭제합니다. 코어 계정의 패치 소스 리포지토리 EC2 인스턴스에 액세스하기 위한 VPC 피어링 연결을 만듭니다. 두 계정의 경로 테이블을 업데이트합니다.
- D.** 포트 80에서 인바운드 트래픽을 차단하는 네트워크 ACL을 만듭니다. 네트워크 ACL을 애플리케이션 계정의 모든 서브넷과 연결합니다. 코어 계정의 패치 소스 리포지토리 EC2 인스턴스에 액세스하기 위한 전송 게이트웨이를 만듭니다. 두 계정의 경로 테이블을 업데이트합니다.

해설

정답: C

-요구사항:

EC2 인스턴스가 인터넷에 접근하지 않아야 한다.

EC2 인스턴스가 Amazon S3, AWS Systems Manager, 그리고 코어 계정의 개인 VPC에 있는 패치 소스 저장소에 접근할 수 있어야 한다.

-솔루션:

VPC 엔드포인트를 통해 Systems Manager와 Amazon S3에 사설로 접근할 수 있습니다.

NAT 게이트웨이 삭제는 EC2 인스턴스가 인터넷에 접근하지 못하도록 합니다.

VPC 피어링을 통해 애플리케이션 VPC가 코어 VPC의 패치 소스 저장소에 접근할 수 있습니다.

◆ | Q#0522. | Ref#0522.

미국(US)의 한 회사가 유럽의 한 회사를 인수했습니다. 두 회사 모두 AWS 클라우드를 사용합니다. 미국 회사는 마이크로서비스 아키텍처로 새로운 애플리케이션을 구축했습니다. 미국 회사는 us-east-2 지역의 5개 VPC에서 애플리케이션을 호스팅하고 있습니다. 애플리케이션은 eu-west-1 지역의 한 VPC에 있는 리소스에 액세스할 수 있어야 합니다. 그러나 애플리케이션은 다른 VPC에 액세스할 수 없어야 합니다.

두 지역의 VPC에는 겹치는 CIDR 범위가 없습니다. 모든 계정은 이미 AWS Organizations의 한 조직에 통합되어 있습니다.

어떤 솔루션이 이러한 요구 사항을 가장 비용 효율적으로 충족할까요?

A. eu-west-1에 하나의 트랜짓 게이트웨이를 만듭니다. us-east-2의 VPC와 eu-west-1의 VPC를 트랜짓 게이트웨이에 연결합니다. 각 VPC에 필요한 경로 항목을 만들어 트래픽이 트랜짓 게이트웨이를 통해 라우팅되도록 합니다.

B. 각 지역에 하나의 트랜짓 게이트웨이를 만듭니다. 관련 서브넷을 지역 트랜짓 게이트웨이에 연결합니다. 트래픽이 지역 트랜짓 게이트웨이를 통해 라우팅되도록 각 서브넷에 대한 연관된 경로 테이블에 필요한 경로 항목을 만듭니다. 두 트랜짓 게이트웨이를 피어링합니다.

C. 모든 VPC 간에 풀 메시 VPC 피어링 연결 구성을 만듭니다. 각 VPC에 필요한 경로 항목을 만들어 트래픽이 VPC 피어링 연결을 통해 라우팅되도록 합니다.

D. us-east-2의 각 VPC에 대해 eu-west-1의 VPC로의 VPC 피어링 연결을 하나씩 만듭니다. 각 VPC에 필요한 경로 항목을 만들어 트래픽이 VPC 피어링 연결을 통해 라우팅되도록 합니다.

해설

정답: D

Inter-region VPC 피어링을 사용하여 다른 리전 간 VPC를 피어링 한다.

가장 경제적인 솔루션이다.

◆ | Q#0523. | Ref#0523.

여행사에서 Amazon Simple Email Service(Amazon SES)를 사용하여 사용자에게 이메일 알림을 보내는 웹 애플리케이션을 구축했습니다. 이 회사는 이메일 배달 문제를 해결하는 데 도움이 되는 로깅을 활성화해야 합니다. 또한 수신자, 제목, 보낸 시간을 기준으로 검색을 수행할 수 있는 기능이 필요합니다.

이러한 요구 사항을 충족하기 위해 솔루션 아키텍트는 어떤 단계 조합을 취해야 합니까? (두 가지를 선택하세요.)

A. Amazon Data Firehose를 대상으로 하는 Amazon SES 구성 세트를 만듭니다. 로그를 Amazon S3 버킷으로 보내도록 선택합니다.

B. AWS CloudTrail 로깅을 활성화합니다. 로그의 대상으로 Amazon S3 버킷을 지정합니다.

C. Amazon Athena를 사용하여 Amazon S3 버킷의 로그를 쿼리하여 수신자, 제목, 전송 시간을 확인합니다.

D. Amazon CloudWatch 로그 그룹을 만듭니다. Amazon SES를 구성하여 로그를 로그 그룹으로 보냅니다.

E. Amazon Athena를 사용하여 Amazon CloudWatch의 로그에서 수신자, 제목, 전송 시간을 쿼리합니다.

해설

정답: A,C

Amazon Firehose로 SES 로그를 S3로 보내 저장한 후 Athena에서 쿼리할 수 있다.

◆ | Q#0524. | Ref#0524.

한 회사가 AWS로 마이그레이션하여 AWS Business Support를 사용합니다. 이 회사는 AWS 계정에서 Amazon EC2 인스턴스의 비용 효율성을 모니터링하려고 합니다. EC2 인스턴스에는 부서, 사업부 및 환경에 대한 태그가 있습니다. 개발 EC2 인스턴스는 비용이 높지만 사용률이 낮습니다.

이 회사는 사용률이 낮은 개발 EC2 인스턴스를 감지하여 중지해야 합니다. 인스턴스는 지난 14일 중 최소 4일 동안 평균 일일 CPU 사용률이 10% 이하이고 네트워크 I/O가 5MB 이하인 경우 사용률이 낮습니다.

어떤 솔루션이 이러한 요구 사항을 가장 적은 운영 오버헤드로 충족할까요?

- A.** 부서, 사업부 및 환경에 대한 태그를 기반으로 EC2 인스턴스 활용도를 모니터링하도록 Amazon CloudWatch 대시보드를 구성합니다. 활용도가 낮은 개발 EC2 인스턴스를 중지하기 위해 AWS Lambda 함수를 호출하는 Amazon EventBridge 규칙을 만듭니다.
- B.** AWS Systems Manager를 구성하여 EC2 인스턴스 활용도를 추적하고 활용도가 낮은 인스턴스를 Amazon CloudWatch에 보고합니다. 부서, 사업부 및 환경에 대한 태그로 CloudWatch 데이터를 필터링합니다. 활용도가 낮은 개발 EC2 인스턴스를 중지하기 위해 AWS Lambda 함수를 호출하는 Amazon EventBridge 규칙을 만듭니다.
- C.** AWS Trusted Advisor에서 보고한 EC2 인스턴스의 낮은 사용률을 감지하기 위한 Amazon EventBridge 규칙을 만듭니다. 부서, 사업부 및 환경에 대한 태그로 데이터를 필터링하고 사용률이 낮은 개발 EC2 인스턴스를 중지하는 AWS Lambda 함수를 호출하도록 규칙을 구성합니다.
- D.** 모든 EC2 인스턴스의 사용률 데이터를 검색하기 위해 매일 실행하는 AWS Lambda 함수를 만듭니다. 데이터를 Amazon DynamoDB 테이블에 저장합니다. DynamoDB 테이블을 데이터 소스로 사용하여 활용도가 낮은 개발 EC2 인스턴스를 식별하고 중지하는 Amazon QuickSight 대시보드를 만듭니다.

해설

정답: C

Trusted Advisor의 자동화된 보고서를 활용하여 유지관리를 최소화할 수 있다.

◆ | Q#0525. | Ref#0525.

한 회사가 향후 3년 동안 운영될 프로젝트를 위해 AWS에서 애플리케이션을 호스팅하고 있습니다. 이 애플리케이션은 네트워크 로드 밸런서(NLB)의 대상 그룹에 등록된 20개의 Amazon EC2 온디맨드 인스턴스로 구성되어 있습니다. 인스턴스는 두 개의 가용 영역에 분산되어 있습니다. 이 애플리케이션은 상태 비저장이며 주 7일, 하루 24시간 운영됩니다.

이 회사는 애플리케이션에서 느린 응답을 경험하는 사용자로부터 보고서를 받습니다. 성능 지표에 따르면 인스턴스는 일반적인 애플리케이션 사용 시 CPU 사용률이 10%입니다. 그러나 CPU 사용률은 일반적으로 몇 시간 동안 지속되는 바쁜 시간에 100%로 증가합니다.

이 회사는 애플리케이션의 느린 응답 문제를 해결하기 위해 새로운 아키텍처가 필요합니다.

이러한 요구 사항을 가장 비용 효율적으로 충족할 솔루션은 무엇입니까?

- A.** 자동 확장 그룹을 만듭니다. 자동 확장 그룹을 NLB의 대상 그룹에 연결합니다. 최소 용량을 20으로, 원하는 용량을 28로 설정합니다. 20개 인스턴스에 대한 예약 인스턴스를 구매합니다.
- B.** 요청 유형이 request인 Spot Fleet을 만듭니다. TotalTargetCapacity 매개변수를 20으로 설정합니다. DefaultTargetCapacityType 매개변수를 On-Demand로 설정합니다. Spot Fleet을 만들 때 NLB를 지정합니다.
- C.** 요청 유형이 유지 관리인 Spot Fleet을 만듭니다. TotalTargetCapacity 매개변수를 20으로 설정합니다. DefaultTargetCapacityType 매개변수를 Spot으로 설정합니다. NLB를 Application Load Balancer로 교체합니다.

D. 자동 확장 그룹을 만듭니다. 자동 확장 그룹을 NLB의 대상 그룹에 연결합니다. 최소 용량을 4로, 최대 용량을 28로 설정합니다. 4개 인스턴스에 대한 예약 인스턴스를 구매합니다.

해설

정답: D

일반적인 애플리케이션 사용 시 인스턴스의 CPU 사용률이 10% 이므로 최소 4개의 용량이 필요하며, 이는 최소한의 구성이자 가장 비용 효율적인 구성입니다.

◆ | Q#0526. | Ref#0526.

Accompany는 공장에서 센서 데이터를 수집하고 전송하는 애플리케이션을 구축하고 있습니다. 이 애플리케이션은 AWS IoT Core를 사용하여 수백 대의 기기에서 Amazon S3 데이터 레이크로 데이터를 전송합니다. 이 회사는 Amazon S3에 데이터를 로드하기 전에 데이터를 보강해야 합니다.

이 애플리케이션은 5초마다 센서 데이터를 전송합니다. 애플리케이션이 데이터를 수집한 후 30분 이내에 Amazon S3에서 새 센서 데이터를 사용할 수 있어야 합니다. 다른 애플리케이션은 AWS IoT Core에서 센서 데이터를 처리하지 않습니다.

이러한 요구 사항을 가장 비용 효율적으로 충족하는 솔루션은 무엇입니까?

- A. AWS IoT Core에서 센서 데이터를 수집하기 위한 토픽을 만듭니다. 데이터를 풍부하게 하고 Amazon S3에 데이터를 쓰기 위한 AWS Lambda 함수를 만듭니다. Lambda 함수를 호출하기 위한 AWS IoT 규칙 작업을 구성합니다.
- B. AWS IoT Core Basic Ingest를 사용하여 센서 데이터를 수집합니다. AWS IoT 규칙 작업을 구성하여 Amazon Kinesis Data Firehose에 데이터를 씁니다. Kinesis Data Firehose 버퍼링 간격을 900초로 설정합니다. Kinesis Data Firehose를 사용하여 AWS Lambda 함수를 호출하여 데이터를 풍부하게 합니다. Kinesis Data Firehose를 구성하여 데이터를 Amazon S3에 전달합니다.
- C. AWS IoT Core에서 센서 데이터를 수집하기 위한 토픽을 만듭니다. 데이터를 Amazon Timestream 테이블로 보내도록 AWS IoT 규칙 작업을 구성합니다. Timestream에서 데이터를 읽는 AWS Lambda 함수를 만듭니다. 데이터를 풍부하게 하고 Amazon S3에 데이터를 쓰도록 Lambda 함수를 구성합니다.
- D. AWS IoT Core Basic Ingest를 사용하여 센서 데이터를 수집합니다. AWS IoT 규칙 작업을 구성하여 Amazon Kinesis Data Streams에 데이터를 씁니다. 소비자 AWS Lambda 함수를 만들어 Kinesis Data Streams의 데이터를 처리하고 데이터를 풍부하게 합니다. Lambda 함수에서 S3 PutObject API 작업을 호출하여 Amazon S3에 데이터를 씁니다.

해설

정답: B

비용 효율적: IoT Core Basic Ingest는 대용량 저가치 데이터에 가장 비용 효율적인 옵션입니다.
낮은 지연 시간: 버퍼링 간격이 900초인 Kinesis Data Firehose는 비용과 지연 시간 간의 균형을 제공하여 30분 이내에 S3에서 데이터를 사용할 수 있다는 요구 사항을 충족합니다.
확장성: Kinesis Data Firehose는 높은 처리량을 처리할 수 있으므로 대량의 센서 데이터에 적합합니다.
단순성: 이 솔루션에는 최소한의 구성 요소가 있는 간단한 파이프라인이 포함됩니다.

◆ | Q#0527. | Ref#0527.

한 회사가 대규모 IoT 기기에서 데이터를 수집하고 있습니다. 이 데이터는 Amazon S3 데이터 레이크에 저장됩니다. 데이터 과학자는 별도의 AWS 계정에서 VPC의 두 퍼블릭 서브넷에서 실행되는 Amazon EC2 인스턴스에서 분석을 수행합니다.

데이터 과학자는 EC2 인스턴스에서 데이터 레이크에 액세스해야 합니다. EC2 인스턴스에는 이미 Amazon S3에 액세스할 수 있는 권한이 있는 역할이 할당되어 있습니다. 회사 정책에 따라 권한이 있는 네트워크만 IoT 데이터에 액세스할 수 있습니다.

솔루션 아키텍트는 이러한 요구 사항을 충족하기 위해 어떤 단계 조합을 취해야 하나요? (두 가지를 선택하세요.)

- A.** 데이터 과학자의 VPC에 Amazon S3에 대한 게이트웨이 VPC 엔드포인트를 생성합니다.
- B.** 데이터 과학자의 AWS 계정에서 데이터 레이크에 대한 S3 액세스 포인트를 만듭니다.
- C.** EC2 인스턴스 역할을 업데이트합니다. s3:DataAccessPointArn 조건 키의 값이 유효한 액세스 포인트 ARN일 때 s3:GetObject 작업을 허용하는 조건이 있는 정책을 추가합니다.
- D.** S3 트래픽을 S3 액세스 포인트로 라우팅하기 위해 VPC 경로 테이블을 업데이트합니다.
- E.** s3:DataAccessPointArn 조건 키의 값이 유효한 액세스 포인트 ARN인 경우 s3:GetObject 작업을 허용하는 조건이 포함된 S3 버킷 정책을 추가합니다.

해설

정답: B,E

AWS 계정에서 데이터레이크에 대한 S3 액세스 포인트 생성(B) 후 S3 버킷 정책 추가(E)

DataAccessPointArn은 액세스 포인트가 생성되었을 때만 작동한다.

◆ | Q#0528. | Ref#0528.

한 회사가 자사 웹사이트를 AWS로 마이그레이션하려고 합니다. 이 웹사이트는 온프레미스 자체 관리형 Kubernetes 클러스터에 배포된 컨테이너를 사용합니다. 웹사이트의 모든 데이터는 온프레미스 PostgreSQL 데이터베이스에 저장됩니다.

이 회사는 온프레미스 Kubernetes 클러스터를 Amazon Elastic Kubernetes Service(Amazon EKS) 클러스터로 마이그레이션하기로 결정했습니다. EKS 클러스터는 정적 노드 수가 있는 EKS 관리 노드 그룹을 사용합니다. 이 회사는 또한 온프레미스 데이터베이스를 Amazon RDS for PostgreSQL 데이터베이스로 마이그레이션합니다.

솔루션 아키텍트는 마이그레이션 전에 이 워크로드의 총 소유 비용(TCO)을 추정해야 합니다.

어떤 솔루션이 필요한 TCO 정보를 제공할까요?

- A.** Migration Evaluator에 대한 액세스를 요청합니다. Migration Evaluator Collector를 실행하고 데이터를 가져옵니다. 시나리오를 구성합니다. Migration Evaluator에서 Quick Insights 보고서를 내보냅니다.
- B.** 온프레미스 데이터베이스에 대한 AWS Database Migration Service(AWS DMS)를 시작합니다. 평가 보고서를 생성합니다. AWS 가격 계산기에서 EKS 마이그레이션 비용에 대한 추정치를 만듭니다.
- C.** AWS 애플리케이션 마이그레이션 서비스를 초기화합니다. 온프레미스 서버를 소스 서버로 추가합니다. 테스트 인스턴스를 시작합니다. 애플리케이션 마이그레이션 서비스에서 TCO 보고서를 출력합니다.
- D.** AWS Cloud Economics Center 웹페이지에 액세스하여 AWS Cloud Value Framework를 평가합니다. Cloud Value Framework에서 AWS 비용 및 사용 보고서를 만듭니다.

해설

정답: A

포괄적인 TCO 분석: Migration Evaluator는 마이그레이션 프로젝트를 평가하도록 특별히 설계되었으며 자세한 비용 견적을 제공합니다.

정확한 데이터: 온프레미스 환경에서 데이터를 수집함으로써 Migration Evaluator는 보다 정확한 비용 견적을 생성할 수 있습니다.

시나리오 모델링: 시나리오를 구성하는 기능을 통해 다양한 마이그레이션 옵션과 관련 비용을 테스트할 수 있습니다.

Quick Insights 보고서: 이는 잠재적 TCO에 대한 요약된 개요를 제공합니다.

◆ | Q#0529. | Ref#0529.

이벤트 회사가 AWS에서 티켓팅 플랫폼을 운영합니다. 회사의 고객은 플랫폼에서 이벤트를 구성하고 예약합니다. 이벤트로 인해 플랫폼으로의 트래픽이 크게 증가합니다. 회사는 각 고객의 이벤트 날짜와 시간을 알고 있습니다.

회사는 Amazon Elastic Container Service(Amazon ECS) 클러스터에서 플랫폼을 운영합니다. ECS 클러스터는 자동 확장 그룹에 있는 Amazon EC2 주문형 인스턴스로 구성됩니다. 자동 확장 그룹은 예측적 확장 정책을 사용합니다.

ECS 클러스터는 Amazon S3 버킷에 자주 요청하여 티켓 자산을 다운로드합니다. ECS 클러스터와 S3 버킷은 동일한 AWS 리전과 동일한 AWS 계정에 있습니다. ECS 클러스터와 S3 버킷 간의 트래픽은 NAT 게이트웨이를 통해 흐릅니다.

회사는 플랫폼의 가용성을 저하시키지 않고 플랫폼 비용을 최적화해야 합니다.

이러한 요구 사항을 충족하는 단계 조합은 무엇입니까? (두 가지를 선택하십시오.)

- A.** S3 버킷에 대한 게이트웨이 VPC 엔드포인트를 생성합니다.
- B.** Spot Instances의 Auto Scaling 그룹을 사용하는 다른 ECS 용량 공급자를 추가합니다. 새 용량 공급자 전략이 기존 용량 공급자 전략과 동일한 가중치를 갖도록 구성합니다.
- C.** 예약된 스케일링 정책의 기간 동안 해당 인스턴스 유형에 대한 주문형 용량 예약을 생성합니다.
- D.** S3 버킷에서 S3 전송 가속을 활성화합니다.
- E.** 예약된 이벤트에 대한 예측적 확장 정책을 예약된 확장 정책으로 교체합니다.

해설

정답: A,E

동일 리전의 S3 통신의 경우 게이트웨이 VPC 엔드포인트를 사용하면 통신 요금이 면제된다.(A)

이벤트 날짜와 시간을 알고있으면 예약된 스케일링 정책을 사용하여 비용을 절감할 수 있다.(E)