

1) 한 회사가 여러 가용 영역에 걸쳐 VPC에서 퍼블릭 3티어 웹 애플리케이션을 실행합니다. 프라이빗 서브넷에서 실행되는 애플리케이션 티어의 Amazon EC2 인스턴스는 인터넷에서 소프트웨어 패치를 다운로드해야 합니다. 하지만 EC2 인스턴스는 인터넷에서 직접 액세스할 수 없습니다.

EC2 인스턴스가 필요한 패치를 다운로드할 수 있도록 하려면 어떤 작업을 수행해야 하나? (2 개 선택)

- A) 퍼블릭 서브넷에서 NAT 게이트웨이를 구성합니다.
- B) 인터넷 트래픽에 대한 NAT 게이트웨이 경로가 있는 사용자 지정 라우팅 테이블을 정의하고 이를 애플리케이션 티어의 프라이빗 서브넷과 연결합니다.
- C) EC2 인스턴스에 탄력적 IP 주소를 할당합니다.
- D) 인터넷 트래픽에 대한 인터넷 게이트웨이 경로가 있는 사용자 지정 라우팅 테이블을 정의하고 이를 애플리케이션 티어의 프라이빗 서브넷과 연결합니다.
- E) 프라이빗 서브넷에서 NAT 인스턴스를 구성합니다.

2) 한 솔루션 아키텍트가 2 주간의 회사 폐쇄 기간 중에 실행할 필요가 없는 Amazon EC2 인스턴스의 비용을 절감하기 위한 솔루션을 설계하려고 합니다. EC2 인스턴스에서 실행되는 애플리케이션은 인스턴스가 작동을 재개할 때 존재해야 하는 인스턴스 메모리에 데이터를 저장합니다.

이 솔루션 아키텍트는 EC2 인스턴스를 종료하고 재개하기 위해 어떤 접근 방식을 제안해야 하나?

- A) 인스턴스 스토어 볼륨에 데이터를 저장하도록 애플리케이션을 수정합니다. EC2 인스턴스를 다시 시작하는 동안 볼륨을 다시 연결합니다.
- B) EC2 인스턴스를 중지하기 전에 EC2 인스턴스의 스냅샷을 생성합니다. 인스턴스를 다시 시작한 후 스냅샷을 복원합니다.
- C) 절전 모드가 활성화된 EC2 인스턴스에서 애플리케이션을 실행합니다. 2 주 동안 회사를 폐쇄하기 전에 인스턴스를 절전 모드로 전환합니다.
- D) EC2 인스턴스를 중지하기 전에 각 EC2 인스턴스의 가용 영역을 기록해 둡니다. 2 주간의 회사 폐쇄가 끝나면 동일한 가용 영역에서 인스턴스를 다시 시작합니다.

3) 한 회사가 VPC 에서 Amazon EC2 인스턴스에 대한 모니터링 애플리케이션을 실행할 계획입니다. 인스턴스의 프라이빗 IPv4 주소를 사용하여 EC2 인스턴스에 연결됩니다. 솔루션 아키텍트는 애플리케이션에 장애가 발생하여 연결할 수 없는 경우 트래픽을 대기 EC2 인스턴스로 신속하게 보낼 수 있는 솔루션을 설계해야 합니다.

이러한 요구 사항을 충족할 수 있는 방법은 무엇입니까?

- A) 사설 IP 주소에 대한 리스너로 구성된 Application Load Balancer 를 배포하고 로드 밸런서에 프라이머리 EC2 인스턴스를 등록합니다. 실패 시 인스턴스를 등록 취소하고 대기 EC2 인스턴스를 등록합니다.
- B) 사용자 지정 DHCP 옵션 세트를 구성합니다. 프라이머리 EC2 인스턴스에 장애가 발생할 경우 대기 EC2 인스턴스에 동일한 사설 IP 주소를 할당하도록 DHCP 를 구성합니다.
- C) 사설 IP 주소로 구성된 EC2 인스턴스에 세컨더리 탄력적 네트워크 인터페이스를 연결합니다. 프라이머리 EC2 인스턴스에 연결할 수 없는 경우 네트워크 인터페이스를 대기 EC2 인스턴스로 이동합니다.
- D) 탄력적 IP 주소를 프라이머리 EC2 인스턴스의 네트워크 인터페이스와 연결합니다. 실패 시 프라이머리 인스턴스에서 탄력적 IP 를 분리하고 이를 대기 EC2 인스턴스와 연결합니다.

4) 한 분석 회사가 사용자에게 웹 분석 서비스를 제공할 계획입니다. 이 서비스를 사용하면 사용자의 웹페이지에 회사의 Amazon S3 버킷에 인증된 GET 요청을 보내는 JavaScript 스크립트를 포함해야 합니다.

솔루션 아키텍트는 스크립트가 성공적으로 실행되도록 하기 위해 무엇을 해야 할까요?

- A) S3 버킷에 교차 출처 리소스 공유(CORS)를 활성화합니다.
- B) S3 버킷에 S3 버전 관리를 활성화합니다.
- C) 사용자에게 스크립트의 서명된 URL 을 제공합니다.
- D) 퍼블릭 실행 권한을 허용하도록 S3 버킷 정책을 구성합니다.

5) 한 회사의 보안 팀이 클라우드에 저장된 모든 데이터를 온프레미스에 저장된 암호화 키를 사용하여 항상 암호화된 상태로 저장할 것을 요구합니다.

이러한 요구 사항을 충족하는 암호화 옵션은 무엇입니까? (2 개 선택)

- A) Amazon S3 관리형 암호화 키를 이용하는 서버 측 암호화(SSE-S3)를 사용합니다.
- B) AWS KMS 관리형 암호화 키를 이용하는 서버 측 암호화(SSE-KMS)를 사용합니다.
- C) 고객 제공 암호화 키를 이용하는 서버 측 암호화(SSE-C)를 사용합니다.
- D) 미사용 데이터 암호화를 제공하기 위해 클라이언트 측 암호화를 사용합니다.
- E) 고객의 키를 사용하여 데이터를 암호화하기 위해 Amazon S3 이벤트에 의해 호출된 AWS Lambda 함수를 사용합니다.

6) 한 회사가 데이터 처리 워크로드를 실행하기 위해 Amazon EC2 예약형 인스턴스를 사용합니다. 야간 작업은 일반적으로 실행하는 데 7 시간이 걸리며 10 시간 내에 완료되어야 합니다. 이 회사는 매월 말에 일시적인 수요 증가로 인해 현재 리소스 용량으로는 시간 제한을 초과하여 작업이 실행될 것으로 예상합니다. 처리 작업은 시작되고 나면 완료 전에 중단할 수 없습니다. 이 회사는 가능한 한 비용 효율적으로 증가된 리소스 용량을 제공하는 솔루션을 구현하고자 합니다.

솔루션스 아키텍트는 이를 달성하기 위해 무엇을 해야 합니까?

- A) 수요가 많은 기간에 온디맨드 인스턴스를 배포합니다.
- B) 추가 인스턴스에 대한 두 번째 EC2 예약을 생성합니다.
- C) 수요가 많은 기간에 스팟 인스턴스를 배포합니다.
- D) EC2 예약에서 EC2 인스턴스 크기를 늘려 증가된 워크로드를 지원합니다.

7) 한 회사가 주간 라이브 TV 프로그램을 위해 온라인 투표 시스템을 운영합니다. 방송 중에 사용자는 **Auto Scaling** 그룹에서 실행되는 **Amazon EC2** 인스턴스의 프론트 엔드 플릿에 몇 분 내에 수십만 개의 투표를 제출합니다. EC2 인스턴스는 **Amazon RDS** 데이터베이스에 투표를 기록합니다. 그러나 데이터베이스는 EC2 인스턴스에서 오는 요청을 모두 처리할 수 없습니다. 솔루션스 아키텍트는 다운타임 없이 가장 효율적인 방식으로 투표를 처리하는 솔루션을 설계해야 합니다.

이러한 요구 사항을 충족하는 솔루션은 무엇입니까?

- A) 프론트 엔드 애플리케이션을 AWS Lambda 로 마이그레이션합니다. Amazon API Gateway 를 사용하여 사용자 요청을 Lambda 함수로 라우팅합니다.
- B) 데이터베이스를 다중 AZ 배포로 변환하여 수평적으로 크기를 조정합니다. 프라이머리 DB 인스턴스와 세컨더리 DB 인스턴스 모두에 기록하도록 프론트 엔드 애플리케이션을 구성합니다.
- C) Amazon Simple Queue Service(Amazon SQS) 대기열에 투표를 보내도록 프론트 엔드 애플리케이션을 구성합니다. SQS 대기열을 읽고 데이터베이스에 투표 정보를 기록하도록 작업자 인스턴스를 프로비저닝합니다.
- D) Amazon EventBridge(Amazon CloudWatch Events)를 통해 예약된 이벤트를 생성하여 투표 기간 동안 더 큰 메모리 최적화 인스턴스로 데이터베이스를 다시 프로비저닝합니다. 투표가 종료되면 더 작은 인스턴스를 사용하도록 데이터베이스를 다시 프로비저닝합니다.

8) 한 회사가 퍼블릭 및 프라이빗 서브넷에서 실행되는 2 티어 애플리케이션 아키텍처를 보유하고 있습니다. 웹 애플리케이션을 실행하는 Amazon EC2 인스턴스는 퍼블릭 서브넷에 있고 데이터베이스의 EC2 인스턴스는 프라이빗 서브넷에서 실행됩니다. 웹 애플리케이션 인스턴스와 데이터베이스는 단일 가용 영역(AZ)에서 실행됩니다.

이 아키텍처에 고가용성을 제공하기 위해 솔루션스 아키텍트가 수행해야 하는 단계는 무엇입니까? (2 개 선택)

- A) 동일한 AZ 에 퍼블릭 서브넷과 프라이빗 서브넷을 새로 생성합니다.
- B) 웹 애플리케이션 인스턴스에 대해 다중 AZ 에 걸쳐 Amazon EC2 Auto Scaling 그룹 및 Application Load Balancer 를 생성합니다.
- C) 기존 웹 애플리케이션 인스턴스를 Application Load Balancer 뒤에 있는 Auto Scaling 그룹에 추가합니다.
- D) 새 AZ 에 퍼블릭 서브넷과 프라이빗 서브넷을 새로 생성합니다. 새 AZ 의 퍼블릭 서브넷에 있는 EC2 인스턴스를 사용하여 데이터베이스를 생성합니다. 이전 데이터베이스 콘텐츠를 새 데이터베이스로 마이그레이션합니다.
- E) 동일한 VPC 에서 퍼블릭 서브넷과 프라이빗 서브넷을 각각 새 AZ 에 새로 생성합니다. 프라이빗 서브넷에서 Amazon RDS 다중 AZ DB 인스턴스를 생성합니다. 이전 데이터베이스 콘텐츠를 새 DB 인스턴스로 마이그레이션합니다.

9) 한 웹사이트에서 매일 정오에 급증하는 트래픽을 수신하는 사용자 지정 웹 애플리케이션을 실행합니다. 사용자는 매일 새 사진과 콘텐츠를 업로드하지만 시간 초과에 대해 불평하고 있습니다. 아키텍처는 Amazon EC2 Auto Scaling 그룹을 사용하며, 애플리케이션은 부팅 시 사용자 요청에 응답하기 전에 시작하는 데 1 분이 걸립니다.

솔루션스 아키텍트는 변화하는 트래픽에 더 잘 대응하기 위해 아키텍처를 어떻게 재설계해야 합니까?

- A) 느린 시작 구성으로 Network Load Balancer 를 구성합니다.
- B) EC2 인스턴스의 직접 요청을 오프로드하도록 Amazon ElastiCache for Redis 를 구성합니다.
- C) EC2 인스턴스 워업 조건을 사용하여 Auto Scaling 단계별 크기 조정 정책을 구성합니다.
- D) Application Load Balancer 를 출처로 사용하도록 Amazon CloudFront 를 구성합니다.

10) AWS 에서 실행되는 한 애플리케이션이 데이터베이스에 Amazon Aurora 다중 AZ DB 클러스터 배포를 사용합니다. 성능 지표를 평가할 때 솔루션스 아키텍트는 데이터베이스 읽기로 인해 I/O 가 크게 발생하고 데이터베이스에 대한 쓰기 요청에 대기 시간이 추가된다는 사실을 발견했습니다.

솔루션스 아키텍트는 읽기 요청과 쓰기 요청을 분리하기 위해 무엇을 해야 할까요?

- A) Aurora 데이터베이스에 연속 읽기 캐싱을 활성화합니다.
- B) 다중 AZ 대기 인스턴스에서 읽기를 수행하도록 애플리케이션을 업데이트합니다.
- C) 적절한 엔드포인트를 사용하도록 Aurora 복제본을 생성하고 애플리케이션을 수정합니다.
- D) 두 번째 Aurora 데이터베이스를 생성하고 프라이머리 데이터베이스에 읽기 전용 복제본으로 연결합니다.

정답

- 1) A, B – [NAT 게이트웨이](#)는 프라이빗 서브넷의 EC2 인스턴스에서 인터넷 또는 기타 AWS 서비스로 트래픽을 전달한 다음 인스턴스에 다시 응답을 보냅니다. NAT 게이트웨이가 생성된 후 인터넷 트래픽이 NAT 게이트웨이를 가리키도록 프라이빗 서브넷의 라우팅 테이블을 업데이트해야 합니다.
- 2) C – EC2 인스턴스를 [절전 모드로 전환](#)하면 인스턴스 메모리의 콘텐츠가 Amazon Elastic Block Store(Amazon EBS) 루트 볼륨에 저장됩니다. 인스턴스가 다시 시작되면 인스턴스 메모리 콘텐츠가 다시 로드됩니다.
- 3) C – [세컨더리 탄력적 네트워크 인터페이스](#)를 EC2 인스턴스에 추가할 수 있습니다. 프라이머리 네트워크 인터페이스는 인스턴스에서 분리할 수 없지만 세컨더리 네트워크 인터페이스는 분리하여 다른 EC2 인스턴스에 연결할 수 있습니다.
- 4) A – 웹 브라우저는 웹페이지와 다른 도메인 이름을 가진 서버에서 시작되는 스크립트 실행을 차단합니다. 스크립트 실행을 허용하는 HTTP 헤더를 전송하도록 [CORS 를 사용하여 Amazon S3 를 구성할 수 있습니다](#).
- 5) C, D – [고객 제공 키를 이용하는 서버 측 암호화\(SSE-C\)](#)를 사용하면 Amazon S3 가 PUT 요청에 제공된 암호화 키를 사용하여 서버 측 객체를 암호화할 수 있습니다. Amazon S3 가 객체의 암호를 해독할 수 있도록 GET 요청에 동일한 키를 제공해야 합니다. 또한 고객은 Amazon S3 에 데이터를 업로드하기 전에 클라이언트 측에서 데이터를 암호화할 수 있으며, 그런 다음 데이터를 다운로드한 후에 데이터의 암호를 해독할 수 있습니다. AWS software development kits(SDKs)는 프로세스를 간소화하는 S3 암호화 클라이언트를 제공합니다.
- 6) A – [스팟 인스턴스](#)는 비용이 가장 적게 드는 옵션이지만 중단할 수 없거나 특정 기간 내에 완료해야 하는 작업에는 적합하지 않습니다. [온디맨드 인스턴스](#)는 실행 중인 시간(초) 만큼 비용이 청구됩니다.
- 7) C – 데이터베이스에서 투표 데이터 흐름을 [분리](#)하면 투표 시스템이 데이터베이스 쓰기를 기다리지 않고 투표를 계속 처리할 수 있습니다. [SQS 대기열](#)에서 읽을 전용 작업자를 추가하면 제어 가능한 속도로 데이터베이스에 투표를 입력할 수 있습니다. 투표는 데이터베이스가 처리할 수 있을 만큼의 속도로 데이터베이스에 추가되며 손실되지 않습니다.

8) B, E – 새 가용 영역(AZ)에 새 서브넷을 생성하여 중복 네트워크를 제공합니다. [로드 밸런서 뒤에 있는 두 AZ에 인스턴스가 포함된 Auto Scaling 그룹](#)을 생성하여 웹 애플리케이션의 고가용성을 보장하고 두 퍼블릭 AZ 간에 웹 트래픽을 재배포합니다. 두 프라이빗 서브넷에 RDS DB 인스턴스를 생성하여 [데이터베이스 티어도 고가용성으로 만듭니다](#).

9) C – 현재 구성에서는 새 EC2 인스턴스가 트랜잭션에 응답하기 전에 새 EC2 인스턴스를 사용하기 시작합니다. 이로 인해 인스턴스가 과도하게 조정될 수도 있습니다. [단계별 크기 조정 정책](#)을 사용하면 새로 시작된 인스턴스가 [웜 업](#)되는 데 걸리는 시간(초)을 지정할 수 있습니다. 웜 업 시간이 만료될 때까지 EC2 인스턴스는 Auto Scaling 그룹의 집계된 지표에 포함되지 않습니다. 스케일 아웃 중에는 Auto Scaling 로직은 웜 업 중인 EC2 인스턴스를 Auto Scaling 그룹의 현재 용량으로 간주하지 않습니다. 따라서 여러 개의 경보 위반이 동일한 단계 조절 범위에 해당하는 경우 단일 크기 조정 작업이 이루어집니다. 이를 통해 필요 이상으로 인스턴스를 추가하지 않게 됩니다.

10) C – [Aurora 복제본](#)은 읽기 트래픽을 오프로드하는 방법을 제공합니다. Aurora 복제본은 주 데이터베이스와 동일한 [기본 스토리지](#)를 공유하므로 일반적으로 지연 시간이 매우 짧습니다. Aurora 복제본에는 자체 엔드포인트가 있으므로 읽기 트래픽을 새 엔드포인트로 보내도록 애플리케이션을 구성해야 합니다.