인프라 이벤트 준비 상태

AWS 지침 및 모범 사례

2017년 7월

고지 사항

이 문서는 정보 제공 목적으로만 제공됩니다. 본 문서의 발행일 당시 AWS의 현재 제품 및 실행방법을 설명하며, 예고 없이 변경될 수 있습니다. 고객은 본 문서에 포함된 정보나 AWS 제품 또는 서비스의 사용을 독립적으로 평가할 책임이 있으며, 각 정보 및 제품은 명시적이든 묵시적이든 어떠한 종류의 보증 없이 "있는 그대로" 제공됩니다. 본 문서는 AWS, 그 계열사, 공급업체 또는 라이선스 제공자로부터 어떠한 보증, 표현, 계약 약속, 조건 또는 보증을 구성하지 않습니다. 고객에 대한 AWS의 책임 및 의무는 AWS 계약에 준거합니다. 본 문서는 AWS와 고객 간의 어떠한 계약도 구성하지 않으며 이를 변경하지도 않습니다.

목차

요약

이 백서에서는 프로덕션 워크로드를 Amazon Web Services(AWS)에 배포한 고객이 제품 출시 또는 절기 트래픽 스파이크 등의 계획된 조정 이벤트를 동적인 방식으로 순조롭게 처리하는 클라우드 기반 애플리케이션을 설계 및 프로비저닝하는 방법에 대한 지침과 모범 사례를 고객에게 설명합니다. 또한, 인프라 이벤트 계획의 여러 개념적 영역에 대한 일반 설계 원칙을 다루고 구체적인 모범 사례 및 지침을 제공합니다. 그런 다음 운영 준비 상태에 대한 고려 사항과 관행 및 이벤트 후 작업에 대해 설명합니다.

서론

인프라 이벤트 준비 상태는 비즈니스에 영향을 미칠 수 있는 중요 예상 이벤트를 설계하고 준비하는 것과 관련되어 있습니다. 이러한 이벤트가 진행되는 동안에는 모든 조건과 변화하는 트래픽 패턴에 대해 회사 웹 서비스가 안정적이고 응답성을 유지하며 고도의 내결함성을 가지는 것이 필수입니다. 이러한 이벤트에는 새로운 지역으로의 확장, 신규 제품 또는 기능 출시, 계절별 이벤트 또는 중요한 비즈니스 발표 또는 마케팅 이벤트 등이 포함될 수 있습니다.

제대로 계획되지 않은 인프라 이벤트는 회사의 비즈니스 평판, 연속성 또는 재무에 악영향을 미칠 수 있습니다. 인프라 이벤트 장애는 예기치 않은 서비스 장애, 부하 관련 성능 저하, 네트워크 지연 시간, 스토리지 용량 제한, API 호출 속도와 같은 시스템 제한, 사용 가능한 IP 주소의 수량 한정, 모니터링 부족으로 인한 애플리케이션 스택 구성 요소 행동에 대한 이해 부족, 타사 서비스 또는 확장 준비가 되지 않은 구성 요소에 대한 예기치 않은 종속성 또는 그 외의 예기치 않은 오류 조건과 같은 형태로 발생할 수 있습니다.

중요한 이벤트 도중에 예기치 않은 장애가 발생할 위험을 최소화하려면 기업은 시간과 리소스를 투자하여 이벤트를 계획 및 준비하고, 직원을 교육하고, 관련 프로세스를 설계 및 문서화해야 합니다. 특정 클라우드 기반 애플리케이션을 위한 인프라 이벤트 계획에 소요되는 투자 규모는 시스템의 복잡성 및 글로벌 배포 범위에 따라 다를 수 있습니다. 이 백서에 제공된 설계 원칙 및 모범 사례 지침은 기업의 클라우드 구축 범위 또는 복잡성에 관계없이 동일하게 적용됩니다.

Amazon Web Services(AWS)를 사용하는 기업은 계획된 조정 이벤트에 대비하여 동적이고 유연한 종량 과금제 방식으로 인프라를 확장할 수 있습니다. 탄력적이고 프로그래밍 가능한 Amazon의 다양한 제품과 서비스는 기업에게 Amazon 자체의 글로벌 네트워크에 사용되는 것과 같은 뛰어난 보안과 신뢰성 및 빠른 속도의 인프라를 제공하며 기업이 빠르게 변화하는 비즈니스 요구 사항에 따라 민첩하게 대응할 수 있게 해 줍니다.

본 백서에서는 인프라 이벤트 계획 및 실행을 안내하는 모범 사례와 설계 원칙을 다루며, AWS 서비스를 사용하여 비즈니스 요구에 따른 애플리케이션의 성능 확장을 준비하는 방법을 설명합니다.

인프라 이벤트 준비 상태 계획

이 섹션에서는 계획된 인프라 이벤트의 구성 요소와 그러한 이벤트 과정에 일반적으로 수반되는 활동 유형을 설명합니다.

계획된 인프라 이벤트는 무엇입니까?

*계획된 인프라 이벤트*는 비즈니스 중심의 예기되고 예약된 이벤트로서 해당 이벤트 기간 동안 비즈니스가 응답성과 확장성이 높고 내결함성이 있는 웹 서비스를 유지하는 것이 필수적입니다. 마케팅 캠페인, 회사 사업부와 관련된 뉴스 이벤트, 제품 출시, 지역 확장 또는 그 외에 회사의 웹 기반 애플리케이션 및 기반 인프라에 추가 트래픽을 유발하는 유사 활동 등의 경우 이러한 사항이 요구됩니다.

계획된 인프라 이벤트 도중 어떤 일이 발생할까요?

대부분의 계획된 인프라 이벤트에서 주된 관심사는 높은 트래픽 수요를 충족할 수 있도록 웹 인프라에 용량을 추가할 수 있는 기능입니다. 물리적 컴퓨팅, 스토리지 및 네트워크 리소스로 프로비저닝된 기존 온프레미스 환경에서 회사의 IT 부서는 이론적 최대 피크에 대한 최상의 예상 수치를 기반으로 추가 용량을 프로비저닝해야 합니다. 이 방식은 용량 프로비저닝이 부족할 수 있는 위험을 내포하며 회사에서는 웹 서버의 과부하, 느린 응답 시간 및 기타 런타임 오류로 인해 비즈니스 손실이 발생할 수 있습니다.

AWS 클라우드는 탄력적이고 프로그래밍 가능한 인프라를 제공합니다. 이는 곧 실시간 수요에 맞추어 신속한 프로비저닝이 가능함을 의미합니다. 또한 시스템 지표에 대하여 지능적이고 동적인 자동화된 방식으로 응답하여 웹 서버 클러스터, 프로비저닝된 처리 능력, 스토리지 용량, 사용 가능한 컴퓨팅 코어, 스트리밍 샤드 수 등의 리소스가 필요에 따라 확장 또는 축소되도록 구성할 수 있습니다.

그 뿐 아니라 많은 AWS 서비스는 완전관리형으로 제공됩니다. 이러한 서비스에는 스토리지, 데이터베이스, 분석, 애플리케이션 및 배포 서비스가 포함됩니다. 따라서 AWS 고객은 트래픽이 많은 이벤트를 위하여 이러한 서비스를 구성할 때의 복잡성을 염려할 필요가 없습니다. AWS의 완전관리형 서비스는 확장성과 고가용성을 제공하도록 설계되었습니다.

일반적으로, 계획된 인프라 이벤트를 준비하기 위해 AWS 고객은 시스템 검토를 수행하여 애플리케이션 아키텍처 및 운영 준비 상태를 평가하고 확장성 및

내결함성을 고려합니다. 정상적인 비즈니스 활동 성능과 비교된 트래픽 예상치가 고려되고 용량 지표 및 필요한 추가 용량 예상치가 산출됩니다. 병목 현상을 일으킬 수 있는 모든 요소와 타사 업스트림 및 다운스트림 종속성이 식별되고 적절히 해결됩니다. 계획된 이벤트가 지역 확장 또는 신규 사용자 집단의 유입을 포함하는 경우 지리적 요소도 고려됩니다. 다른 AWS 리전 또는 가용 영역으로의 확장은 계획된 이벤트 전에 수행됩니다. Auto Scaling, 로드 밸런싱, 지역 라우팅, 고가용성, 장애 조치와 같은 고객의 AWS 동적 시스템 설정에 대한 검토도 수행되어 이러한 설정이 예상 볼륨 및 트랜잭션 속도 증가를 처리할 수 있도록 올바르게 구성되어 있는지 확인합니다. AWS 리소스 제한 및 콘텐츠 전송 네트워크(CDN) 오리진 서버의 위치와 같은 정적 설정도 고려되고 필요에 따라 수정됩니다.

또한, 모니터링 및 알림 메커니즘을 검토하여 이벤트에 대한 실시간 투명성을 제공하고 계획된 이벤트 완료 후 사후 분석을 제공할 수 있도록 개선합니다.

문제 해결이 필요하거나 서버 중단과 같이 실시간 지원이 필요한 경우 AWS 고객은 계획된 이벤트 동안 AWS에 지원 사례를 열 수도 있습니다. AWS Enterprise Support 플랜에 가입한 고객에게는 지원 엔지니어와 즉각적으로 통화하거나 신속한 응답이 필요한 경우 중요한 심각도 사례를 제출할 수 있는 유연성이 추가적으로 제공됩니다.

AWS 리소스는 이벤트 후 트래픽 수준에 맞게 자동 축소되거나 이벤트의 상태에 따라 계속 확장 가능하도록 설계되어 있습니다.

설계 원칙

계획된 이벤트에 대한 준비는 클라우드 기반 애플리케이션 스택 또는 워크로드 구현의 최초 시점부터 좋은 설계를 함으로써 시작됩니다.

개별화된 워크로드

좋은 설계는 정상 및 높아진 트래픽 수준 모두에서 계획된 이벤트 워크로드를 효과적으로 관리하는 데 필수적인 요소입니다. 처음부터 특정 비즈니스 애플리케이션 또는 제품을 중심으로 개별화된 독립적 리소스 기능 그룹을 설계해야 합니다. 이 섹션에서는 이러한 설계 목표에 대해 다양한 관점에서 설명합니다.

태깅

태그는 리소스에 레이블을 지정하고 구성하는 데 사용됩니다. 태그는 계획된 인프라 이벤트 동안 인프라 리소스를 관리하는 데 필요한 핵심 인프라 구성 요소입니다. AWS에서 태그는 로드 밸런서 또는 Amazon Elastic Compute Cloud(EC2) 인스턴스와 같이 관리되는 개별 리소스에 적용되며 고객이 관리하는 키 값 레이블입니다. 잘 정의되어 AWS 리소스에 첨부된 태그를 참조하면 전체 인프라에서 어느 리소스가 계획된 이벤트 워크로드를 구성하는지 쉽게 확인할 수 있습니다. 그런 다음 이 정보를 사용하여 준비 태세를 위한 분석 작업을 수행할 수 있습니다. 비용 할당을 위해서도 태그를 사용할 수 있습니다.

예를 들어 태그는 EC2 인스턴스, Amazon 머신 이미지(AMI) 이미지, 로드 밸런서, 보안 그룹, Amazon Relational Database Service(RDS) 리소스, Amazon Virtual Private Cloud(VPC) 리소스, Amazon Route 53 상태 확인 및 Amazon Simple Storage Service(S3) 버킷을 구성하는 데 사용될 수 있습니다.

효과적인 태깅 전략에 대한 자세한 내용은 AWS 태깅 전략[1]을 참조하십시오.

태그를 생성 및 관리하고 리소스 그룹에 포함시키는 방법에 대한 예는 AWS를 위한 리소스 그룹 및 태깅[2]을 참조하십시오.

소결합

클라우드를 위한 설계를 할 때는 애플리케이션 스택의 모든 구성 요소를 가능한 한 서로 독립적으로 작동하도록 설계해야 합니다. 그러면 클라우드 기반 워크로드에 복원력 및 확장성이 증가합니다.

클라우드 기반 애플리케이션 스택의 각 구성 요소를 잘 정의된 입력 및 출력용 인터페이스(예: RESTful API)를 가진 블랙 박스로 설계하면 구성 요소 간의 상호 종속성을 줄일 수 있습니다. 구성 요소가 애플리케이션이 아니고 함께 애플리케이션을 구성하는 서비스인 경우 이를 *마이크로 서비스 아키텍처*라고 부릅니다. 애플리케이션 구성 요소 간의 통신 및 조정을 위해 그림 1에 표시된 AWS 메시지 대기열과 같은 이벤트 중심의 알림 메커니즘을 사용하여 구성 요소 간에 메시지를 전달할 수 있습니다.

그림 1. RESTful 인터페이스와 메시지 대기열을 사용한 소결합

이러한 메커니즘을 사용하면 구성 요소 중 하나의 변경 또는 장애가 다른 구성 요소로 전파될 가능성이 대폭 줄어듭니다. 예를 들어, 멀티 티어 애플리케이션 스택의 서버 중 하나가 응답하지 않을 경우 소결합된 애플리케이션이 응답하지 않는 티어를 건너뛰거나 성능 저하 모드의 대체 트랜잭션으로 전환하도록 설계할 수 있습니다.

또한 중간 메시지 대기열을 사용하는 소결합된 애플리케이션 구성 요소는 보다 쉽게 비동기 통합을 위해 설계될 수 있습니다. 애플리케이션의 구성 요소가 지점 간 직접 통신을 사용하지 않고 대신 중간 및 영구 메시징 계층(예: Amazon Simple Queue Service(SQS) 대기열 또는 Amazon Kinesis Streams와 같은 스트리밍 데이터 메커니즘)을 사용하므로 다운스트림 구성 요소가 수신 대기열을 처리하는 동안 다른 구성 요소의 갑작스러운 활동 증가를 수용할 수 있습니다. 또는

구성요소 장애가 발생하는 경우, 장애 구성 요소가 복구될 때까지 메시지가 대기열 또는 스트림에서 지속될 수 있습니다.

AWS에서 제공하는 메시지 대기열 및 알림 서비스에 대한 자세한 내용은 Amazon Simple Queue Service(SQS)[3]를 참조하십시오.

서버가 아닌 서비스

관리형 서비스 및 서비스 엔드포인트를 사용하면 보안, 액세스, 백업, 복원, 패치 관리, 변경 관리, 모니터링 또는 보고 설정에 대해 걱정할 필요가 없으며 기존의 수많은 시스템 관리에 대한 세부 사항도 신경쓸 필요가 없습니다. 이러한 클라우드 리소스는 고가용성 및 복원성을 위해 다중 가용 영역(또는 일부의 경우 다중 리전) 구성을 사용하여 사전에 프로비저닝될 수 있습니다. 이는 많은 경우 다운타임 없이 확장 또는 축소 가능하며 AWS Management Console 또는 API/CLI 호출을 통해 즉석에서 구성할 수 있습니다.

관리형 서비스 및 서비스 엔드포인트는 관계형 및 NoSQL 데이터베이스 시스템, 데이터 웨어하우징, 이벤트 알림, 객체 및 파일 스토리지, 실시간 스트리밍, 빅 데이터 분석, 기계 학습, 검색, 트랜스코딩 및 기타 다양한 기능을 포함하는 고객 애플리케이션 스택을 지원하는 데 사용할 수 있습니다. 엔드포인트는 AWS 서비스의 진입점이 되는 URL입니다. 예를 들어 https://dynamodb.us-west-2.amazonaws.com은 Amazon DynamoDB 서비스의 진입점입니다.

관리형 서비스 및 해당 서비스 엔드포인트를 사용하면 계획된 인프라 이벤트 동안 증가되는 볼륨, 서비스 범위 및 트랜잭션 속도를 처리하기 위한 설계 솔루션의 일부로 프로덕션용 리소스를 활용할 수 있습니다. 관리형 서비스와 동일한 기능을 수행하는 자체 서버를 프로비저닝 하고 관리할 필요도 없습니다.

AWS 서비스 엔드포인트에 대한 자세한 내용은 AWS 리전 및 엔드포인트[4]를 참조하십시오. 또한 Amazon EMR[5], Amazon RDS[6] 및 Amazon ECS[7]에서 엔드포인트를 포함한 관리형 서비스의 예를 참조하십시오.

서버 없는 아키텍처

계획된 인프라 이벤트 동안 동적으로 변화하는 프로세싱 로드에 응답하는 문제를 효율적으로 처리할 수 있는 또 하나의 전략은 AWS Lambda를 활용하는 것입니다. Lambda는 이벤트 중심의 서버 없는 컴퓨팅 플랫폼입니다. 동적으로 호출되는 이 서비스는 (알림을 통한) 이벤트에 대한 응답으로 Python, Node.js 또는 Java 코드를 실행하고 해당 코드에 명시된 컴퓨팅 리소스를 자동 관리합니다. Lambda는 Amazon EC2 컴퓨팅 리소스의 사전 프로비저닝을 필요로

하지 않습니다. Amazon Simple Notification Service(Amazon SNS)에서 Lambda 함수를 트리거하도록 구성할 수 있습니다. Amazon SNS에 대한 자세한 내용은 Amazon Push Notification Service[8]를 참조하십시오.

서버 없는 Lambda 함수는 데이터베이스 운영, 데이터 변환, 객체 또는 파일 검색뿐 아니라 외부 이벤트 또는 내부 시스템 부하 지표에 따른 조정 작업과 같은 다른 AWS 서비스를 액세스 또는 호출하는 코드를 실행할 수 있습니다. 또한 AWS Lambda는 새 알림 또는 자체 이벤트를 생성하고 다른 Lambda 함수를 실행할 수도 있습니다.

AWS Lambda를 사용하면 계획된 인프라 이벤트 동안 조정 작업을 정교하게 제어할 수 있습니다. 예를 들어, Lambda는 타사 시스템에 조정 작업이 필요함을 알리거나 프로비저닝되는 새 인스턴스에 네트워크 인터페이스를 추가하는 등 Auto Scaling 작업의 기능을 확장하는 데 사용할 수 있습니다. Lambda를 사용하여 조정 작업을 사용자 지정하는 방법의 예는 AWS Lambda의 Auto Scaling 수명 주기 후크 사용[9]을 참조하십시오.

AWS Lambda에 대한 자세한 내용은 AWS Lambda란 무엇입니까?[10]를 참조하십시오.

자동화

Auto Scaling

인프라 이벤트 계획의 중요한 구성 요소 중 하나는 Auto Scaling입니다. 사전 정의된 조건에 따라 애플리케이션 용량의 확장 또는 축소를 자동 조정하는 기능은 계획된 인프라 이벤트에서 트래픽 패턴 및 볼륨이 변동하는 동안 애플리케이션 가용성을 유지하는 데 도움이 됩니다.

AWS는 EC2 인스턴스, 데이터베이스 용량, 컨테이너 등 다양한 리소스에 대한 Auto Scaling 기능을 제공합니다.

클라우드 기반 애플리케이션을 구성하는 서버 플릿 등 인스턴스 그룹의 조정에 Auto Scaling을 사용하면 특정 조건을 기반으로 한 자동 조정이 가능합니다. 또한 Auto Scaling은 특정 인스턴스가 비정상 상태가 될 경우에도 고정된 수의 인스턴스를 유지하는 데에도 사용할 수 있습니다. 이 자동 조정 및 인스턴스 수 유지 기능은 Auto Scaling 서비스의 핵심 기능입니다.

Auto Scaling은 그룹 내 인스턴스에 대한 주기적인 상태 확인을 수행하여 사용자가 지정한 인스턴스 수를 유지합니다. 인스턴스가 비정상 상태가 되면 그룹은 비정상 인스턴스를 종료하고 이를 대체할 다른 인스턴스를 시작합니다.

Auto Scaling 정책을 사용하여 변화하는 조건을 충족하도록 서버 그룹에서 실행되는 EC2 인스턴스 수를 자동으로 증가 또는 감소할 수 있습니다. 조정 정책이 적용되면 Auto Scaling 그룹은 그룹의 용량을 원하는 대로 조절하고 필요에 따라 인스턴스를 시작 또는 종료합니다. 조정 작업은 동적으로 실행될 수 있고 또는 트래픽의 등락과 흐름이 알려져 있거나 예측 가능한 경우 일정에 따라 실행될 수 있습니다.

재시작 및 복구

모든 계획된 인프라 이벤트의 중요한 설계 요소 중 하나는 손상된 인스턴스 또는 서버를 처리하고 즉시 복구 또는 재시작할 수 있는 절차 및 자동화를 구축하는 것입니다.

EC2 인스턴스는 기반 하드웨어에 대한 시스템 상태 확인이 실패할 때 자동 복구되도록 설정할 수 있습니다. 인스턴스는 재부팅(필요한 경우 새 하드웨어 사용)되지만 해당 인스턴스 ID, IP 주소, 탄력적 IP 주소, Amazon Elastic Block Store(EBS) 볼륨 첨부 및 기타 구성 세부 정보는 그대로 유지됩니다. EC2 인스턴스의 자동 복구에 대한 자세한 내용은 Amazon EC2의 자동 복구[11]를 참조하십시오.

구성 관리/오케스트레이션

강력하고 안정적이며 신속하게 반응하는 계획된 인프라 이벤트 전략의 필수 요소 중 하나는 개별 리소스 상태 관리 및 애플리케이션 스택 배포를 위한 구성 관리 및 오케스트레이션 도구의 통합입니다.

일반적으로 구성 관리 도구는 서버 인스턴스, 로드 밸런서, Auto Scaling, 개별 애플리케이션 배포 및 애플리케이션 상태 모니터링의 프로비저닝 및 구성을 처리합니다. 이러한 도구는 또한 데이터베이스, 스토리지 볼륨, 캐싱 계층과 같은 추가 서비스를 통합하는 기능도 제공합니다.

구성 관리 위에 위치하는 추상화 계층인 오케스트레이션 도구는 이러한 다양한 리소스의 관계를 지정할 수 있는 수단을 제공함으로써 고객이 리소스 종속성을 걱정할 필요 없이 복수의 리소스를 하나의 통합 클라우드 애플리케이션 인프라로 프로비저닝 및 관리할 수 있도록 합니다.

이러한 도구는 개별 리소스뿐 아니라 해당 리소스 간의 관계도 코드 형태로 정의 및 설명하기 때문에 이 코드에 대한 버전 관리를 통해 이전 버전으로 롤백하거나 테스트 및 개발 용도로 새로운 코드 분기를 시도할 수 있습니다. 또한 인프라 이벤트에 최적화된 오케스트레이션 및 구성을 정의하고 이벤트 후에 표준 구성으로 롤백할 수 있습니다.

Amazon Web Services는 코드형 하드웨어 배포 및 오케스트레이션의 달성을 위해 다음과 같은 도구를 권장합니다.

- **AWS Config 및 Config Rules** 또는 AWS Config 파트너 - AWS 리소스, 구성 기록 및 리소스 구성 규정 준수에 대한 상세하며 검색 가능한 시각적 인벤토리를 제공합니다.
- **AWS CloudFormation** 또는 타사 AWS 리소스 오케스트레이션 도구 - AWS 리소스 프로비저닝, 업데이트 및 종료를 관리합니다.
- **AWS OpsWorks, Elastic Beanstalk** 또는 타사 서버 구성 관리 도구 - 운영 체제(OS) 및 애플리케이션 구성 변경을 관리합니다.

코드형 하드웨어를 관리하는 방법에 대한 자세한 내용은 인프라 구성 관리를 참조하십시오.[12]

다양성/복원력

단일 장애 지점 및 병목 현상 제거

인프라 이벤트에 대한 계획을 수립할 때 모든 애플리케이션 스택에서 단일 장애 지점(SPOF)또는 성능 병목 현상이 있는지 분석해야 합니다. 장애가 발생했을 경우 전체 또는 상당한 부분의 애플리케이션의 작동을 중지시킬 수 있는 단일의 서버, 데이터 볼륨, 데이터베이스 NAT 게이트웨이 또는 로브 밸런서 인스턴스가 있습니까?

두 번째로, 클라우드 기반 애플리케이션이 트래픽 또는 트랜잭션 볼륨에 따라 확장될 때 해당 데이터 흐름 경로를 따라 데이터 볼륨이 증가하면서 네트워크 대역폭, CPU 프로세싱 주기와 같은 물리적 한도 또는 제한에 직면할 수 있는 인프라 구성 요소가 있습니까?

이러한 위험을 식별했으면 다양한 방법으로 문제를 완화할 수 있습니다.

고장 대비 설계

앞서 언급된 것처럼, 소결합 및 메시지 대기열을 RESTful 인터페이스와 함께 사용하는 것은 개별 리소스 장애 또는 변동하는 트래픽이나 트랜잭션 볼륨에 대비한 복원력을 달성하는 데 유용한 전략입니다. 복원력이 뛰어난 설계의 또 다른 차원은 애플리케이션 구성 요소를 최대한 상태 비저장(stateless)이 되도록 구성하는 것입니다.

상태 비저장 애플리케이션은 이전 트랜잭션에 대한 지식을 필요로 하지 않으며 다른 애플리케이션 구성 요소에 대한 종속성이 느슨하게 구성되어 있습니다. 또한 어떠한 세션 정보도 저장하지 않습니다. 풀 또는 클러스터 내의 아무 인스턴스나 요청을 처리할 수 있으므로, 상태 비저장 애플리케이션은 풀 또는 클러스터의 구성원으로 수평 확장될 수 있습니다. Auto Scaling 및 상태 확인 기준을 사용하여 변동하는 컴퓨팅, 용량 및 처리량 요구 사항을 프로그래밍 방식으로 처리하도록 하면 필요에 따라 더 많은 리소스를 간단하게 추가할 수 있습니다. 애플리케이션을 상태 비저장으로 설계한 후에는 EC2 인스턴스 대신 Lambda 함수를 사용하여 서버 없는 아키텍처에 리팩터링할 수 있습니다. Lambda 함수는 동적 조정 기능도 내장하고 있습니다.

웹 서버와 같은 애플리케이션 리소스가 트랜잭션에 대한 상태 저장 데이터를 가지는 것을 피할 수 없는 상황에서는 애플리케이션에서 상태 저장 데이터를 가지는 부분이 서버 자체로부터 분리되도록 설계할 것을 고려해야 합니다. 예를 들어, HTTP 쿠키 또는 이에 상응하는 상태 데이터를 DynamoDB와 같은 데이터베이스 또는 S3 버킷 또는 EBS 볼륨에 저장할 수 있습니다.

복잡한 다단계 워크플로를 사용하고 있으며 각 워크플로 단계의 현재 상태를 추적할 필요가 있는 경우 Amazon Simple Workflow Service(SWF)를 사용하여 실행 기록을 중앙에 저장하고 이러한 워크로드를 상태 비저장으로 만들 수 있습니다.

또 다른 복원력 조치로는 분산 프로세싱을 채택하는 방법이 있습니다. 적시에 대용량 데이터를 처리해야 하지만 단일 컴퓨팅 리소스로 이러한 요구 사항을 충족할 수 없는 사용 사례의 경우, 작업 및 데이터가 더 작은 조각으로 분할되어 컴퓨팅 리소스 클러스터 전반에 걸쳐 병렬로 실행되도록 워크로드를 설계할 수 있습니다. 분할된 데이터 및 작업이 처리되는 독립 노드가 실패할 수 있으므로 분산 프로세싱은 상태가 저장되지 않습니다. 이 경우 실패한 작업을 분산 프로세싱 클러스터의 다른 노드에서 자동 재시작하는 작업은 분산 프로세싱 일정 엔진으로 자동 처리합니다.

AWS는 Amazon EMR, Amazon Athena 및 Amazon Machine Learning과 같은 다양한 분산 데이터 프로세싱 엔진을 제공합니다. 이러한 엔진은 모두 엔드포인트를 제공하고 패칭, 유지 관리, 조정, 장애 조치 등의 복잡성을 제거해 주는 관리형 서비스입니다.

스트리밍 데이터를 실시간으로 처리하는 경우, Amazon Kinesis Streams를 사용하여 데이터를 여러 샤드로 분할하면 이를 Lambda 함수 또는 EC2 인스턴스와 같은 해당 데이터의 여러 소비자가 처리할 수 있습니다.

이러한 유형의 워크로드에 대한 자세한 내용은 AWS의 빅 데이터 분석 옵션[13]을 참조하십시오.

다중 영역 및 다중 리전

AWS 서비스는 전 세계 여러 곳에서 호스팅되고 있습니다. 이러한 위치는 리전과 가용 영역으로 구성됩니다. 리전은 분리된 지리적 영역입니다. 각 리전은 여러 개의 격리된 위치를 가지며 이를 가용 영역이라고 합니다. AWS는 고객에게 인스턴스와 같은 리소스 및 데이터를 여러 위치에 배치할 수 있는 기능을 제공합니다.

애플리케이션은 복수의 가용 영역 및 리전에 걸쳐 배포되도록 설계해야 합니다. 가용 영역 및 리전 전반에 걸친 리소스 배포 및 복제와 더불어, 로드 밸런싱 및 장애 조치 메커니즘을 사용하여 장애가 발생했을 때 애플리케이션 스택이 자동으로 데이터 흐름 및 트래픽을 이러한 대체 위치로 경로를 재지정하도록 애플리케이션을 설계해야 합니다.

로드 밸런싱

Elastic Load Balancing(ELB) 서비스를 사용하면, 애플리케이션 서버 플릿을 로드 밸런서에 연결하면서도 여러 가용 영역에 걸쳐 배포할 수 있습니다. 로드 밸런서 뒤에 자리한 특정 가용 영역에 배치된 EC2 인스턴스의 상태 확인에 실패하는 경우 로드 밸런서는 해당 노드로의 트래픽 전송을 중단합니다. Auto Scaling을 함께 사용하면 정상 노드 수가 다른 가용 영역과 자동으로 다시 밸런싱되며 수동으로 개입할 필요가 없습니다.

또한 Amazon Route 53 및 지연 시간 기반 DNS 라우팅 알고리즘을 사용하면 리전 전반에 걸친 로드 밸런싱을 구현할 수 있습니다. 자세한 내용은 지연 시간 기반 라우팅을 참조하십시오.[14]

로드 쉐딩 전략

클라우드 기반 인프라에서 *로드 쉐딩*의 개념은 기본 시스템의 부담을 완화하기 위해 트래픽을 다른 곳으로 리디렉션하거나 프록시하는 과정으로 구성됩니다. 경우에 따라 로드 쉐딩 전략은 프로세싱 부하를 감소하여 최소한 수신 요청의 일부라도 처리할 수 있도록 특정 트래픽 스트림을 삭제하거나 애플리케이션 기능을 줄이는 분류 작업이 될 수 있습니다.

로드 쉐딩에는 다양한 기술을 사용할 수 있습니다. 지연 시간 기반 DNS 라우팅이 그 중 한 방법입니다. 또 다른 방법으로는 캐시 사용이 있습니다. Amazon ElastiCache와 같은 인 메모리 캐싱 계층을 사용하면 애플리케이션 가까이에서 캐싱을 수행할 수 있습니다. 또는 Amazon CloudFront와 같은 글로벌 콘텐츠 배포 네트워크를 사용하여 사용자의 엣지에 가까운 캐싱 계층을 사용할 수 있습니다.

ElastiCache 및 CloudFront에 대한 자세한 내용은 ElastiCache[15] 및 Amazon CloudFront CDN[16] 시작하기를 참조하십시오.

비용 최적화

예약, 스팟 및 온 디맨드

클라우드에서 시스템 지표, 기타 성능 및 상태 확인 기준을 기반으로 리소스를 동적으로 프로비저닝하는 기능과 긴밀하게 연결된 기능은 클라우드의 리소스 프로비저닝 비용을 제어하는 기능입니다. Auto Scaling을 사용하면 리소스 사용률을 실제 프로세싱 및 스토리지 요구에 밀접하게 일치시켜 비용 낭비와 사용률이 낮은 리소스를 최소화할 수 있습니다.

클라우드에서는 온디맨드 인스턴스, 예약 인스턴스(RI) 또는 스팟 인스턴스 중에서 선택할 수 있는 기능을 사용하여 비용을 제어할 수 있습니다. 또한, DynamoDB를 위한 예약 용량 기능도 사용할 수 있습니다.

온디맨드 인스턴스에서는 사용한 EC2 인스턴스에 대해서만 지불하면 됩니다. 온디맨드 인스턴스를 사용하면 장기 약정 없이 시간당 컴퓨팅 파워 사용량에 따라 요금을 지불할 수 있습니다.

Amazon EC2 예약 인스턴스는 온디맨드 인스턴스 요금과 비교하여 상당한 할인 혜택(최대 75%)을 제공하며 특정 가용 영역에서 사용할 때 용량을 예약할 수 있습니다. 그러나 가용성 예약 및 결제액 할인 외에는 예약 인스턴스와 온 디맨드 인스턴스 사이에 기능 차이가 없습니다.

스팟 인스턴스를 사용하면 예비 Amazon EC2 컴퓨팅 용량에 입찰할 수 있습니다. 스팟 인스턴스는 온디맨드 요금과 비교하여 할인된 요금으로 사용할 수 있을 때가 많으므로 클라우드 기반 애플리케이션의 실행 비용을 크게 줄일 수 있습니다.

클라우드용 시스템을 설계할 때 일부 사용 사례는 스팟 인스턴스를 사용하는 것이 더 적합할 수 있습니다. 예를 들어, 스팟 인스턴스는 입찰 가격이 사용자의 입찰 가격을 초과하면 언제든지 폐기될 수 있으므로 비교적 상태를 저장하지 않고 수평 확장되는 애플리케이션 스택에서만 실행할 것을 고려해야 합니다. 상태를 저장하는 애플리케이션이나 값비싼 프로세싱 부하의 경우 예약 인스턴스 또는 온디맨드 인스턴스가 더 나을 수 있습니다. 용량 제한을 전혀 고려할 수 없는 미션 크리티컬 애플리케이션의 경우 예약 인스턴스가 최적의 선택입니다.

자세한 내용은 예약 인스턴스[17] 및 스팟 인스턴스[18] 참조하십시오.

이벤트 관리 프로세스

인프라 이벤트 계획은 애플리케이션 개발자, 관리자 및 비즈니스 이해 관계자가 관여되는 그룹 활동입니다. 인프라 이벤트의 수주 전부터 웹 서비스의 각 주요 인프라 구성 요소를 소유 및 운영하는 주요 기술 직원이 참여하는 회의를 정기적으로 반복해야 합니다.

인프라 이벤트 일정

인프라 이벤트 계획은 이벤트 날짜 수주 전부터 시작되어야 합니다. 그림 2는 계획된 이벤트 수명 주기의 일반적인 타임라인입니다.

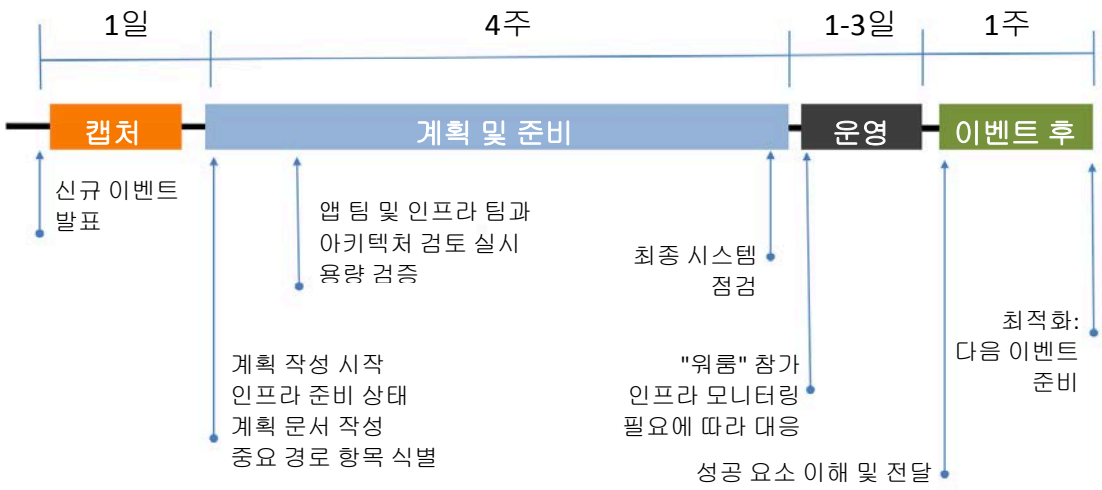

그림 2. 일반적인 인프라 이벤트 타임라인

계획 및 준비

일정

인프라 이벤트 전 몇 주 동안 다음과 같은 활동 일정이 권장됩니다.

제 1주:

- 인프라 이벤트의 계획 및 엔지니어링을 추진할 팀을 지정합니다.
- 이해 관계자 간의 회의를 소집하여 이벤트의 파라미터(규모, 기간, 시간, 지리적 범위, 영향을 받는 워크로드) 및 성공 기준을 이해합니다.
- 모든 다운스트림 또는 업스트림 파트너 및 협력 공급업체를 참여시킵니다.

제 2-3주:

- 아키텍처를 검토하고 필요에 따라 조절합니다.
- 운영 검토를 실시하고 필요에 따라 조절합니다.
- 이 백서와 주석 참조에서 설명한 모범 사례를 따릅니다.
- 위험을 식별하고 완화 계획을 개발합니다.
- 계획된 이벤트 실행서를 개발합니다.

제 4주:

- 예상 부하를 기반으로 조정이 필요한 모든 클라우드 공급업체 서비스를 검토합니다.
- 서비스 한도를 확인하고 필요에 따라 한도를 늘립니다.
- 모니터링 대시보드 및 지정한 임계값에 대한 알림을 설정합니다.

아키텍처 검토

인프라 이벤트 준비의 필수 부분 중 하나는 트래픽 증가를 경험할 애플리케이션 스택의 아키텍처 검토입니다. 검토의 목적은 애플리케이션 확장성 또는 안정성에 대한 잠재적 위험을 확인 및 식별하고 이벤트 전에 최적화할 기회를 식별하는 것입니다.

AWS는 Enterprise Support 고객에게 고객 애플리케이션 스택 검토를 위해 5개의 설계 핵심 요소에 중점을 둔 프레임워크를 제공합니다. 이 5가지는 보안, 안정성, 성능 효율성, 비용 최적화 및 운영 효율성입니다(아래 설명 참조).

표 1: 잘 설계된 애플리케이션의 핵심 요소

핵심 요소 이름	핵심 요소 정의	관련 관심 영역
보안	위험 평가 및 완화 전략을 통해 비즈니스 가치를 제공하는 동시에 정보, 시스템 및 자산을 보호하는 기능.	자격 증명 관리, 암호화, 모니터링, 로깅, 키 관리, 전용 인스턴스, 규정 준수, 거버넌스
안정성	시스템이 인프라 또는 서비스 장애를 복구하고, 수요에 맞게 컴퓨팅 리소스를 동적으로 확보하고, 구성 오류 또는 일시적 네트워크 문제와 같은 중단을 완화할 수 있는 기능.	서비스 한도, 여러 가용 영역 및 리전, 확장성, 상태 확인/모니터링, 백업/재해 복구, 네트워킹, 자가 복구 자동화
성능 효율성	시스템 요구 사항을 충족하기 위해 컴퓨팅 리소스를 효율적으로 사용하고 수요 변화와 기술 혁신에 맞추어 효율성을 유지할 수 있는 기능.	올바른 AWS 서비스, 리소스 사용률, 스토리지 아키텍처, 캐싱, 지연 시간 요구 사항
비용 최적화	불필요한 비용 또는 준최적화 상태의 리소스를 피하거나 제거할 수 있는 기능.	스팟/예약 인스턴스, 환경 튜닝, 서비스 선택, 볼륨 튜닝, 계정 관리, 통합 결제, 리소스 작동 중단
운영 효율성	비즈니스 가치를 제공하고 지속적으로 지원 프로세스 및 절차를 개선할 수 있도록 시스템을 실행 및 모니터링하는 기능.	실행서, 플레이북, CI/CD, 게임 데이, 코드형 인프라, RCA

AWS 기반 애플리케이션 스택의 검토에 사용할 수 있는 상세한 아키텍처 검토 항목 체크리스트를 이 백서의 부록에서 사용할 수 있습니다.

운영 검토

애플리케이션의 설계 구성 요소에 중점을 둔 아키텍처 검토 외에도 클라우드 운영 및 관리 방식을 검토하여 클라우드 워크로드 관리를 얼마나 잘 처리하고 있는지 평가해야 합니다. 검토의 목표는 운영 문제를 식별하고 이벤트 전에 조치를 취해 이 문제를 최소화하는 데 있습니다.

AWS는 Enterprise Support 고객에게 인프라 이벤트 준비에 유용한 도구가 될 수 있는 클라우드 운영 검토 기능을 제공합니다. 검토 기능은 다음과 같은 영역의 평가에 중점을 둡니다.

- 준비 태세 – 조직 구조, 프로세스 및 기술의 올바른 조합을 갖추고 있어야 합니다. 애플리케이션 스택을 관리하는 직원의 역할과 책임을 명확하게 정의해야 합니다. 프로세스는 이벤트에 맞춰 미리 정의해야 합니다. 절차는 가능한 한 자동화되어야 합니다.
- 모니터링 – 효과적인 모니터링은 애플리케이션의 성능을 측정합니다. 모니터링은 문제가 발생하기 전에 이상 징후를 감지하는 데 필수적인 기능이며 부정적인 영향을 최소화할 수 있는 기회를 제공합니다.
- 작업 – 운영 활동은 가능한 한 자동화를 활용하여 적시에 안정적인 방식으로 수행되어야 하며 에스컬레이션을 필요로 하는 예기치 않은 운영 이벤트도 처리할 수 있어야 합니다.
- 최적화 – 수집한 지표, 운영 추세 및 학습한 내용을 사용하여 사후 분석을 수행함으로써 미래 이벤트를 위한 개선 기회를 캡처 및 보고합니다. 최적화와 준비 태세를 함께 갖추면 운영 문제를 해결하고 재발을 방지하기 위한 피드백 루프를 만들 수 있습니다.

AWS 서비스 한도 이해

계획된 인프라 이벤트 중에는 애플리케이션 또는 워크로드를 조정하면서 클라우드 공급자가 설정한 서비스 한도를 초과하지 않는 것이 중요합니다.

일반적으로 클라우드 서비스 공급자는 사용자가 사용할 수 있는 서로 다른 리소스를 제한합니다. 보통 이러한 한도는 계정 및 리전별로 적용됩니다. 영향을 받는 리소스로는 인스턴스, 볼륨, 스트림, 서버 없는 호출, 스냅샷, VPC 수, 보안 규칙 등이 있습니다. 이러한 한도는 리소스 남용을 시도하는 악성 코드 또는 불법 행위자를 차단하는 안전 조치이자 결제 위험을 최소화하는 제어 조치 용도로 사용됩니다.

일부 서비스 한도는 시간이 지나 사용자의 클라우드 규모가 확장됨에 따라 자동으로 해제되지만 이러한 서비스 대부분은 사용자가 지원 사례를 열어 한도 증가를 요청해야 합니다. 일부 서비스 한도는 지원 사례를 통해 늘릴 수 있지만 한도를 변경할 수 없는 서비스도 있습니다.

AWS는 Enterprise 및 Business Support 고객에게 사전 예방적으로 모든 서비스 한도를 관리할 수 있도록 한도 점검 대시보드가 포함된 Trusted Advisor를 제공합니다.

다양한 AWS 서비스의 한도와 이를 확인하는 방법에 대한 자세한 내용은 AWS 서비스 한도[19] 및 Trusted Advisor[20]를 참조하십시오.

패턴 이해

기준선

인프라 이벤트를 시작하기 전에 주요 지표에 대한 "정상 복귀" 값을 문서화해야 합니다. 이 값은 이벤트 완료/종료 후 애플리케이션/서비스가 언제 안전하게 정상 수준으로 돌아가는지를 결정하는 데 도움이 됩니다. 예를 들어, 로드 밸런서를 통한 정상 트랜잭션 속도가 초당 요청 2,5000회라고 파악했으면 이벤트 후 정상 복귀 절차를 언제 시작할 지를 결정하는 데 도움이 됩니다.

데이터 흐름 및 종속성

데이터가 애플리케이션의 다양한 구성 요소 사이를 어떻게 흐르는지를 이해하는 것은 잠재적 병목 현상 및 종속성을 식별하는 데 도움이 됩니다. 데이터 흐름에서 데이터 소비자인 애플리케이션 티어 또는 구성 요소가 적합한 크기이며 애플리케이션 스택에서 데이터 생산자인 티어 또는 구성 요소가 확장 가능합니까? 구성 요소 장애 시 해당 구성 요소가 복구될 때까지 데이터를 대기열에 유지할 수 있습니까? 다운스트림 또는 업스트림 데이터 공급자 또는 소비자가 이벤트에 대한 응답으로 확장 가능합니까?

비례성

인프라 이벤트 준비에 있어 검토해야 할 또 다른 고려 사항은 애플리케이션 스택의 다양한 구성 요소에 필요한 확장의 비례성입니다. 비례성은 항상 일대일로 적용되지 않습니다. 예를 들어, 로드 밸런서 전반에 걸친 초당 트랜잭션 수가 10배로 늘어나면 프런트엔드 애플리케이션에서 실행되는 프로세싱 때문에 스토리지 용량 또는 스트리밍 샤드 수 또는 데이터 베이스 읽기 및 쓰기 수를 20배 증가시켜야 할 수 있습니다.

커뮤니케이션 계획

이벤트 전에 커뮤니케이션 계획을 개발해야 합니다. 내부 이해 관계자 및 지원 그룹의 목록을 수집하고 다양한 시나리오의 다양한 이벤트 단계에서 연락할 사람을 식별해야 합니다(예: 이벤트 시작, 이벤트 중, 이벤트 후, 사후 분석, 비상 연락처, 문제 해결 상황에서의 연락처 등).

연락이 필요한 대상으로는 다음과 같은 개인 및 그룹이 포함될 수 있습니다.

- 이해 관계자
- 운영 관리자
- 개발자
- 지원 팀
- 클라우드 서비스 공급업체 팀
- 네트워크 운영 센터(NOC) 팀

내부 연락처 목록을 수집하면서 애플리케이션의 지속적인 라이브 전송에 관련된 외부 이해 관계자의 연락처 목록도 개발해야 합니다. 이러한 이해 관계자에는 스택의 주요 구성 요소를 지원하는 파트너 및 공급업체와 외부 서비스, 데이터 피드, 인증 서비스 등을 제공하는 다운스트림 및 업스트림 공급업체가 포함됩니다.

이 외부 연락처 목록은 다음과 같은 연락처도 포함해야 합니다.

- 인프라 호스팅 공급업체
- 통신 공급업체
- 라이브 데이터 스트리밍 파트너
- PR 마케팅 담당자
- 광고 파트너
- 서비스 엔지니어링에 관련된 기술 컨설턴트

각 공급자에게 다음과 같은 정보를 문의하십시오.

- 이벤트 동안 실시간으로 연락할 수 있는 담당자
- 핵심 지원 연락처 및 에스컬레이션 프로세스

- 이름, 전화 번호 및 이메일 주소
- 라이브 기술 연락처를 사용할 수 있을 것이라는 확인

Enterprise Support를 구독하는 AWS 고객에게는 해당 계정에 기술 계정 관리자(TAM)도 할당되어 전담 AWS 지원 직원이 이벤트를 인식하고 있으며 지원할 준비가 되었음을 조율하고 확인해 줍니다. 또한 TAM은 이벤트 동안 워룸에서 대기하며 필요한 경우 지원 에스컬레이션을 제공합니다.

NOC 준비

이벤트 전에 운영 및/또는 개발자 팀에게 이벤트가 발생했을 때 프로덕션에 사용되는 웹 서비스의 각 핵심 구성 요소를 모니터링할 수 있는 라이브 지표 대시보드를 생성할 것을 지시해야 합니다. 가장 좋은 방법은 이 대시보드가 이벤트 동안 분 단위 또는 그 외의 적절하고 효율적인 간격으로 업데이트된 지표를 자동으로 제공하는 것입니다.

다음과 같은 구성 요소를 모니터링하는 것이 좋습니다.

- 각 서버의 리소스 사용률(CPU, 디스크 및 메모리 사용률)
- 웹 서비스 응답 시간
- 웹 트래픽 지표(사용자, 페이지 보기, 세션)
- 방문자 리전(글로벌 고객 세그먼트)별 웹 트래픽
- 데이터베이스 서버 사용률
- 마케팅 플로우 변환 지표(예: 변환율 및 부수 효과 비율)
- 애플리케이션 오류 로그
- 카나리아 모니터링

Amazon CloudWatch는 CloudWatch 사용자 지정 대시보드를 통해 AWS 리소스로부터 이러한 지표 대부분을 단일 창에 수집할 수 있는 방법을 제공합니다. 또한, CloudWatch는 AWS에서 자동으로 제공하지 않는 사용자 지정 지표를 CloudWatch로 가져올 수 있는 기능을 제공합니다. AWS 모니터링 도구 및 기능에 대한 자세한 내용은 이 백서의 모니터링 섹션을 참조하십시오.

실행서 준비

인프라 이벤트를 준비하기 위한 실행서를 개발해야 합니다. *실행서*는 운영자가 이벤트 동안 수행할 절차와 작업을 모은 운영 매뉴얼입니다. 이벤트 실행서는

일상적인 운영 및 예외 처리에 사용되는 기존 실행서를 바탕으로 작성될 수 있습니다. 일반적으로 실행서는 시스템을 시작, 중지, 감독 및 디버깅하는 절차를 포함합니다. 또한, 예기치 않은 이벤트 및 비상 사태를 처리하는 절차를 설명하고 있어야 합니다.

실행서는 다음과 같은 섹션을 포함해야 합니다.

- **이벤트 세부 정보**: 이벤트, 성공 기준, 언론 보도, 이벤트 날짜 및 고객 측 및 AWS의 주요 이해 관계자의 연락처 정보를 간략히 설명합니다.
- **AWS 서비스 목록**: 이벤트 도중 사용될 모든 AWS 서비스를 열거합니다. 또한 이러한 서비스에 대한 예상 부하, 영향을 받는 리전 및 계정 ID를 열거합니다.
- **아키텍처 및 애플리케이션 검토**: 부하 테스트 결과, 인프라 및 애플리케이션 설계의 스트레스 지점, 워크로드에 대해 측정된 복원력, 단일 장애 지점 및 잠재적 병목 현상을 문서화합니다.
- **운영 검토**: 모니터링 설정, 상태 기준, 알림 메커니즘 및 서비스 복원 절차를 설명합니다.
- **준비 태세 체크리스트**: 서비스 한도 점검, 로드 밸런서와 같은 애플리케이션 스택 구성 요소의 사전 워밍, 스트림 샤드, DynamoDB 파티션, S3 파티션과 같은 리소스 사전 프로비저닝 등의 고려 사항을 포함합니다. 자세한 내용은 이 백서의 부록에 있는 아키텍처 검토 세부 체크리스트를 참조하십시오.

모니터링

모니터링 계획

데이터베이스, 애플리케이션 및 운영 체제 모니터링은 성공적인 이벤트를 위해 매우 중요합니다. 인프라 이벤트 도중 심각한 인시던트를 효과적으로 감지하고 즉각적으로 대응할 수 있는 포괄적인 모니터링 시스템을 구축해야 합니다. 높은 관점에서 볼 때 효과적인 모니터링 전략은 모니터링 도구가 비즈니스 중요도에 따라 애플리케이션에 맞는 적절한 수준의 기능을 갖출 수 있도록 해 줍니다. 효과적인 인시던트 관리 전략은 AWS와 고객 모니터링 데이터 모두를 이벤트 및 인시던트 관리 도구 및 프로세스에 통합합니다. 모든 AWS 솔루션 세그먼트에서 모니터링 데이터를 수집하는 모니터링 계획을 구현하면 복잡한 장애가 발생한 경우 이를 디버깅하는 데 많은 도움이 됩니다.

모니터링 계획을 다음과 같은 질문을 고려해야 합니다.

- 이벤트를 위해 어떤 모니터링 도구와 대시보드를 설정해야 합니까?
- 모니터링 목표와 허용되는 임계값은 무엇입니까? 작업을 트리거하는 이벤트는 무엇입니까?
- 어떤 리소스와 이러한 리소스의 어떤 지표를 모니터링할 것이며 얼마나 자주 폴링을 해야 합니까?
- 모니터링 작업은 누가 수행합니까? 어떤 모니터링 알림이 구현되어 있습니까? 누구에게 알림이 제공됩니까?
- 일반 및 예상 장애에 대해 어떤 해결 계획이 준비되어 있습니까? 예기치 않은 이벤트에 대한 계획은 무엇입니까?
- 장애가 발생한 경우 에스컬레이션 프로세스는 어떻게 됩니까?

이 전략의 일부로 다음과 같은 AWS 모니터링 도구를 사용할 수 있습니다.

- **Amazon CloudWatch**: AWS 대시보드 지표로 바로 사용할 수 있는 솔루션으로서, 모니터링, 알림 및 자동화된 프로비저닝을 제공합니다.
- **Amazon CloudWatch 사용자 지정 지표**: 운영 체제, 애플리케이션 및 비즈니스 지표 수집에 사용됩니다. Amazon CloudWatch API를 사용하면 거의 모든 유형의 사용자 지정 지표를 수집할 수 있습니다.
- **Amazon EC2 인스턴스 상태**: 상태 확인을 보고 자동 재부팅 또는 인스턴스 재시작과 같은 상태 기반의 인스턴스 이벤트를 예약하는 데 사용됩니다.
- **Amazon SNS**: 이벤트 중심의 알림을 설정, 운영 및 전송하는 데 사용됩니다.
- **AWS X-Ray**: 시스템 구성 요소 전반에 걸친 데이터 흐름 분석을 통해 분산 애플리케이션 및 마이크로 서비스 아키텍처의 디버깅 및 분석을 도와 줍니다.
- **Amazon Elasticsearch Service**: 중앙 집중식 로그 수집 및 실시간 로그 분석에 사용됩니다. 휴리스틱한 방식으로 신속하게 문제를 감지할 수 있습니다.
- **타사 도구:** 실시간 분석 및 전체 스택 모니터링 및 가시성 확보에 사용됩니다.
- **표준 운영 체제 모니터링 도구**: OS 수준의 모니터링에 사용됩니다.

AWS 모니터링 도구에 대한 자세한 내용은 자동 및 수동 모니터링[21]을 참조하십시오. 또한 Amazon CloudWatch 대시보드 사용[22] 및 사용자 지정 지표 게시[23]를 참조하십시오.

알림

인프라 이벤트를 위한 설계의 중요한 운영 요소 중 하나는 경보 및 알림의 구성을 모니터링 솔루션에 통합하는 것입니다. 이러한 경보 및 알림은 AWS Lambda와 같은 서비스와 함께 알림 기반 조치를 트리거하는 데 사용될 수 있습니다. 운영 이벤트에 대한 응답을 자동화하는 것은 완화, 롤백 및 복구 작업의 응답성을 극대화하기 위한 핵심 요소입니다.

중앙에서 워크로드를 모니터링하고 주요 운영 지표와 관련된 사용 가능한 로그 및 지표를 기반으로 적절한 경보 및 알림을 생성할 수 있는 도구가 설정되어 있어야 합니다. 여기에는 범위를 초과하는 이상 요소와 서비스 또는 구성 요소 장애가 포함됩니다. 가장 좋은 방법은 저성능 임계값을 초과하거나 장애가 발생한 경우 이러한 알림 및 경보에 대한 응답으로 자가 복구 또는 조정이 자동으로 실행되도록 시스템 아키텍처를 설계하는 것입니다.

앞서 언급되었듯이 AWS는 예기치 않은 운영 이벤트에 대한 응답으로 적절한 경고 및 알림을 생성하고 자동 응답을 가능하게 하는 서비스(Amazon SQS 및 Amazon SNS)를 제공합니다.

운영 준비 상태(이벤트 당일)

계획 실행

이벤트 당일에 인프라 이벤트에 관여하는 핵심 팀은 대시보드를 실시간으로 모니터링하면서 라이브 컨퍼런스 회의에 응답할 준비가 되어 있어야 합니다. 실행서는 개발이 완료되어 사용할 준비가 되어 있어야 합니다. 커뮤니케이션 계획이 잘 정의되어 있고 모든 지원 직원 및 이해 관계자가 이러한 계획을 숙지하고 있어야 하며 비상 사태 계획이 구축되어 있어야 합니다.

워룸

이벤트 동안 다음과 같은 참가자가 참석하는 라이브 컨퍼런스 브리지를 개설해야 합니다.

- 기본 담당 애플리케이션 및 운영 팀
- 운영 팀 리더십

- 기술 제공에 직접 관여한 외부 파트너가 제공하는 기술 리소스
- 비즈니스 이해 관계자

이벤트 기간 동안 이 컨퍼런스 브리지를 통해 진행되는 대화는 최소한으로 유지되어야 합니다. 부정적인 운영 이벤트가 발생할 경우 이벤트에 응답할 수 있는 핵심 담당자가 이미 이 브리지에 상주하면서 조치 및 조언을 제공할 수 있어야 합니다.

리더십 보고

이벤트 동안 주요 리더십 관계자에게 매 시간 이메일을 보냅니다. 이 업데이트는 다음과 같은 내용을 포함해야 합니다.

- 상태 요약: 녹색(정상), 노란색(문제 발생), 빨간색(주요 문제)
- 주요 지표 업데이트
- 발생한 문제, 해결 계획의 상태, 해결 예상 시간
- 워룸 컨퍼런스 브리지의 전화 번호(참석을 원하는 사람이 있는 경우)

이벤트 종료 시 유사한 형식의 최종 요약 이메일을 보내야 합니다.

비상 사태 계획

이벤트 준비 과정의 각 단계에는 테스트 환경에서 검증된 롤백 계획이 있어야 합니다.

롤백 계획을 작성할 때에는 다음과 같은 질문을 고려합니다.

- 이벤트 동안 발생할 수 있는 최악의 시나리오는 무엇입니까?
- 어떤 유형의 이벤트가 홍보에 부정적인 영향을 미칩니까?
- 이벤트 중 장애가 발생할 수 있는 타사 구성 요소 및 서비스는 무엇입니까?
- 어떤 지표를 모니터링해야 문제 시나리오가 진행되고 있음을 알 수 있습니까?
- 각 시나리오의 롤백 계획은 무엇입니까?
- 각 롤백 프로세스는 얼마의 시간이 소요됩니까? 어느 수준의 목표 복구 시점(RPO) 및 목표 복구 시간(RTO)이 허용됩니까? (이러한 개념에 대한 자세한 내용은 AWS를 사용한 재해 복구[24]를 참조하십시오)

다음과 같은 유형의 롤백을 고려하십시오.

- **블루/그린 배포**: 새 프로덕션 애플리케이션 또는 환경을 롤아웃하는 경우 이전 프로덕션 빌드를 온라인으로 유지하여 신속한 복귀가 가능하게 합니다.
- **웜 파일럿**: 필요한 경우 신속하게 확장 가능한 두 번째 리전에 최소한의 환경을 시작합니다. 기본 리전에 장애가 발생하면 백업 리전을 신속하게 확장하고 트래픽을 이 두 번째 리전으로 전환합니다.
- **유지 관리 모드 오류 페이지**: 웹 서비스의 각 계층에서 오류 페이지가 구축되어 트리거 가능한지 확인합니다. 필요에 따라 이러한 오류 페이지에 보다 구체적인 메시지를 추가할 준비가 되어 있어야 합니다.

각 장애 시나리오에 대한 모든 롤백 계획을 테스트하고 문서화합니다.

이벤트 후 활동

사후 분석

사후 분석이 매우 자주 간과되는 이유는 일반적으로 고객이 정상적인 운영 복귀에 너무 초조해하기 때문입니다. 그러나 인프라 이벤트 관리 수명 주기의 일부로 사후 분석을 의무화할 것이 권장됩니다. 사후 분석은 관련된 모든 팀과 협업을 통해 운영 절차, 구현 세부 정보, 장애 조치 및 복구 절차 등 추가 최적화를 필요할 수 있는 영역을 식별합니다. 이 작업은 이벤트 중 애플리케이션 스택에 중단이 발생한 경우에 특히 중요합니다. 또한 이벤트 사후 분석은 근본 원인 분석(RCA) 문서를 개발해야 할 때 관련 문서를 제공하는 데 도움이 됩니다.

정상 복귀 프로세스

인프라 이벤트 완료 직후 정상 복귀 프로세스를 시작해야 합니다. 이 기간 동안 관련 애플리케이션과 서비스를 계속 모니터링하여 트래픽이 정상적인 프로덕션 수준으로 복귀되었는지를 확인하는 것이 좋습니다. 준비 단계에 생성한 모든 상태 대시보드를 사용하여 트래픽 및 트랜잭션 속도의 정상화를 확인합니다. 일부 이벤트는 정상 복귀 기간이 직선적으로 단순하게 진행되지만 어떤 경우에는 고르지 않거나 점진적인 볼륨 감소를 경험할 수 있습니다. 일부 트래픽 패턴은 계속 지속될 수 있습니다. 예를 들어, 급증한 트래픽으로부터의 복귀는 일반적으로 간단한 정상 복귀 절차를 필요로 하지만 새 지리적 리전으로 애플리케이션을 배포 또는 확장한 경우 그 영향이 오래 지속되며 새 트래픽 패턴을 신중하게 모니터링하고 추가 모니터링 작업을 영구 애플리케이션 스택의 일부로 포함해야 합니다.

경우에 따라 이벤트 완료 후 이벤트 관리 작업을 종료하는 것이 안전한지를 결정해야 합니다. 이벤트 완료 또는 종료를 언제 선언할지는 이전에 문서화한 주요 지표의 "정상" 값을 참조합니다. 정상 복귀 활동을 서로 다른 타임라인을 가질 수 있는 두 개의 분기로 분할하는 것이 좋습니다. 첫 번째 분기는 내부 및 외부 이해 관계자에게 커뮤니케이션을 전송하고 서비스 한도를 재설정하는 등 이벤트의 운영 관리에 집중합니다. 두 번째 분기는 축소 절차, 환경 상태 검증 및 설계 변경의 복원 또는 유지 여부를 결정하는 조건 등 정상 복귀의 기술적 측면에 집중합니다.

이러한 분기와 연관된 타임라인은 이벤트의 특성, 주요 지표 및 고객 편의에 따라 다를 수 있습니다. 다음 표에는 각 분기에서 이벤트에 대한 적절한 관리 종료 시점을 결정하는 데 도움이 되는 일반적인 작업이 정리되어 있습니다.

표 2: 운영 정상 복귀 작업

작업	설명
커뮤니케이션	내부 및 외부 이해 관계자에게 이벤트 종료를 알립니다. 커뮤니케이션 종료 시점은 이벤트 완료의 정의에 맞추어져야 합니다. "정상 복귀" 지표를 사용하여 커뮤니케이션을 종료하기에 적절한 시점을 결정합니다. 또는 단계적으로 커뮤니케이션을 종료할 수 있습니다. 예를 들어, 워룸 브리지는 종료하지만 이벤트 후 장애가 발생할 경우에 대비하여 이벤트 에스컬레이션 절차는 그대로 유지할 수 있습니다.
서비스 한도/비용 보존	승격된 서비스 한도를 이벤트 후에도 계속 유지하고 싶은 마음이 생길 수 있으나 서비스 한도는 안전 조치로도 사용된다는 점을 유의해야 합니다. 서비스 한도는 인해 과다한 서비스 사용을 차단하여 문제(예: 계정 손상 또는 자동화 구성 오류)와 비용을 방지합니다.
보고 및 분석	이벤트 지표의 데이터 수집 및 대조, 그리고 패턴, 추세, 문제 영역, 성공적인 절차, 임시 절차, 이벤트 타임라인 및 성공 기준 충족 여부 등의 분석 정보를 개발하고 커뮤니케이션 계획에 식별된 모든 내부 관련자에게 배포합니다. 운영 비용이 이벤트를 지원하는지를 보여주는 자세한 비용 분석도 개발되어야 합니다.
최적화 작업	엔터프라이즈 조직은 운영을 개선하면서 계속 진화합니다. 운영 최적화는 개선 기회를 발견하기 위해 지표, 운영 추세 및 이벤트에서 학습한 내용의 지속적인 수집을 필요로 합니다. 최적화는 준비 과정과 연계되어 운영 문제를 해결하고 재발을 방지하기 위한 피드백 루프를 생성합니다.

표 3: 기술적 정상 복귀 작업

작업	설명
서비스 한도/비용 보존	승격된 서비스 한도를 이벤트 후에도 계속 유지하고 싶은 마음이 생길 수 있으나 서비스 한도는 안전 조치의 역할도 수행한다는 점을 유의해야 합니다. 서비스 한도는 계정 손상으로 인한 악의적 활동 또는 자동화 구성 오류 등을 통한 과다 서비스 사용을 차단하여 운영과 운영 비용을 보호합니다.

작업	설명
축소 절차	준비 단계에서 확장된 리소스를 복원합니다. 이러한 항목은 개별 아키텍처에 따라 달라지지만 다음과 같은 항목이 일반적으로 포함됩니다. EC2/RDS 인스턴스 크기 Auto Scaling 구성 예약 용량 프로비저닝된 IOPS
환경 상태 검증	프로덕션 상태를 기준선 지표와 비교하고 검토하여 이벤트 후 및 축소 후 절차가 완료되었으며 영향을 받은 시스템이 정상 상태를 보고하고 있음을 확인합니다.
아키텍처 변경 처리	이벤트의 특성과 운영 지표에 대한 관찰 결과에 따라 이벤트 준비 과정에 적용한 일부 변경 사항은 그대로 유지할 가치가 있을 수 있습니다. 예를 들어 새 지리적 리전으로의 확장은 해당 리전에 포함된 리소스의 영구 증가를 필요로 하며, DB 파티션 수 또는 볼륨 내 PIOPS 스트림의 샤드 수와 같은 특정 서비스 한도 또는 구성 파라미터의 증가는 그대로 지속할 필요가 있는 성능 튜닝 조치일 수 있습니다.

최적화

인프라 이벤트 관리의 가장 중요한 구성 요소라고 할 수 있는 부분은 이벤트 후 분석 그리고 관찰된 운영 및 아키텍처 문제 및 개선 기회의 식별입니다. 인프라 이벤트는 일회성 이벤트인 경우가 드뭅니다. 이러한 이벤트는 계절마다 반복되거나 특정 애플리케이션의 릴리스에 맞춰지거나 새로운 시장 또는 지역으로 진출하는 회사의 성장에 따른 일부분이 될 수 있습니다. 따라서 모든 인프라 이벤트는 다음 이벤트를 보다 효과적으로 관찰, 개선 및 준비할 수 있는 기회를 제공합니다.

결론

AWS가 탄력적이고 프로그램 가능한 제품 및 서비스의 형태로 제공하는 빌딩 블록을 조합하면 거의 모든 규모의 워크로드를 지원할 수 있습니다. AWS 인프라 이벤트 지침 및 모범 사례와 고가용성의 완전한 서비스를 통해 고객은 주요 비즈니스 이벤트를 설계 및 준비하고 조정 수요를 원활하고 동적인 방식으로 충족하여 신속한 응답과 글로벌 접근성을 보장할 수 있습니다.

기여자

이 문서의 작성에 도움을 준 개인 및 조직은 다음과 같습니다.

- Presley Acuna, AWS 엔터프라이즈 지원 관리자
- Kurt Gray, AWS 글로벌 솔루션 아키텍트

- Michael Bozek, AWS 선임 기술 계정 관리자
- Rovan Omar, AWS 기술 계정 관리자
- Will Badr, AWS 기술 계정 관리자
- Eric Blankenship, AWS 선임 기술 계정 관리자
- Greg Bur, AWS 기술 계정 관리자
- Bill Hesse, AWS 선임 기술 계정 관리자
- Hasan Khan, AWS 선임 기술 계정 관리자
- Varun Bakshi, AWS 선임 기술 계정 관리자

참고 문헌

운영 및 아키텍처 모범 사례에 대한 자세한 내용은 AWS 운영 체크리스트[25]를 참조하십시오. 클라우드 기반 애플리케이션 전송 스택의 평가 작업에 대한 구조적 접근 방식을 보려면 AWS Well Architected Framework[26]를 검토하십시오. AWS에서는 AWS 기술 계정 관리자 및 지원 엔지니어가 이벤트의 설계, 계획 및 이벤트 당일 운영에 보다 직접적으로 참여할 것을 원하는 고객에게 IEM(인프라 이벤트 관리)을 Premium Support 형태로 제공합니다. AWS IEM Premium Support 서비스에 대한 자세한 내용은 인프라 이벤트 관리[27]를 참조하십시오.

부록

세부 아키텍처 검토 체크리스트

예-아니오-해당 없음	보안
□-□-□	AWS 보안 모범 사례를 따라 AWS Identity and Access Management(IAM) 액세스 키와 사용자 암호 및 애플리케이션에 관련된 리소스에 대한 자격 증명을 3개월마다 교체합니다. 모든 계정에 암호 정책을 적용하며 하드웨어 또는 가상 멀티 팩터 인증(MFA) 디바이스를 사용합니다.
□-□-□	IAM을 활용하여 AWS API에 부여되는 고유한 역할 기반의 최소 권한을 제어하는 내부 보안 프로세스 및 컨트롤이 구현되어 있습니다.
□-□-□	포함된 퍼블릭/프라이빗 키 페어를 비롯하여 기밀이거나 민감한 정보를 모두 제거했으며 모든 사용자 지정 Amazon 머신 이미지(AMI)의 모든 SSH 인증 키 파일을 검토했습니다.

예-아니오-해당 없음	보안
□-□-□	AMI 내부에 자격 증명을 포함하는 대신 EC2 인스턴스에 대한 IAM 역할을 사용합니다.
□-□-□	IAM 관리 역할을 생성하고 다른 기능 역할에서 IAM 작업을 제한함으로써 IAM 관리 권한을 정규 사용자 권한과 격리합니다.
□-□-□	EC2 인스턴스에 Windows 또는 Linux 인스턴스용 최신 보안 패치를 적용합니다. Amazon EC2 보안 그룹 규칙, VPC 네트워크 액세스 제어 목록, OS 강화, 호스트 기반 방화벽, 침입 탐지/방지, 모니터링 소프트웨어 구성 및 호스트 인벤토리를 포함한 운영 체제 액세스 제어를 사용합니다.
□-□-□	조직의 AWS에 네트워크 연결이 있으며 기업 환경이 암호화 전송 프로토콜을 사용하는지 확인합니다.
□-□-□	중앙 집중식 로그 및 감사 관리 솔루션을 적용하여 환경에 대한 비정상적 액세스 패턴 또는 악성 공격을 식별 및 분석합니다.
□-□-□	보안 이벤트 및 인시던트 관리, 상호연결 및 보고 프로세스가 구축되어 있습니다.
□-□-□	보안 그룹에 상관없이 어떤 AWS 리소스에도 무제한 액세스가 제공되지 않도록 합니다.
□-□-□	프런트엔드 연결(클라이언트와 로드 밸런서 간)에 대해 보안 프로토콜(HTTPS 또는 SSL), 최신 보안 정책 및 암호화 프로토콜을 사용합니다. 보안 강화를 위해 클라이언트와 로드 밸런서 사이의 요청은 암호화됩니다.
□-□-□	Amazon Route 53 MX 리소스 레코드 세트가 관련 Sender Policy Framework(SPF) 값을 포함하는 TXT 리소스 레코드 세트를 갖도록 구성하여 도메인에서 이메일을 전송할 권한이 있는 서버를 지정합니다.

예-아니오-해당 없음	안정성
□-□-□	Auto Scaling 그룹에 배포된 EC2 인스턴스 플릿에 애플리케이션을 배포하여 사전 정의된 조정 계획에 기반한 자동 수평 확장을 보장합니다. 자세히 알아보기
□-□-□	Auto Scaling 그룹 구성에 Elastic Load Balancing 상태 확인을 사용하여 Auto Scaling 그룹이 해당 EC2 인스턴스의 상태에 따라 작동하도록 합니다. (Auto Scaling 그룹에 로드 밸런서를 사용하는 경우에만 적용)
□-□-□	애플리케이션의 핵심 구성 요소를 여러 가용 영역에 배포하며 데이터를 여러 영역에 적절히 복제합니다. Elastic Load Balancing, Amazon Route 53 또는 적절한 타사 도구를 사용하여 이러한 구성 요소 내의 오류가 애플리케이션의 가용성에 어떤 영향을 미치는지 테스트합니다.
□-□-□	데이터베이스 계층에서 Amazon RDS 인스턴스를 여러 가용 영역에 배포하여 다른 가용 영역에 있는 예비 인스턴스에 동시에 복제함으로써 데이터베이스 가용성을 높입니다.
□-□-□	중단 또는 성능 저하가 발생할 경우 자동 또는 수동 장애 조치를 위한 프로세스가 정의되어 있습니다.
□-□-□	CNAME 레코드를 사용하여 DNS 이름을 서비스에 매핑합니다. A 레코드는 사용하지 않습니다.
□-□-□	Amazon Route 53 레코드 세트에 낮은 TTL(time-to-live) 값을 구성했습니다. 이렇게 하면

예-아니오-해당 없음	안정성
	DNS 확인자가 트래픽 경로를 재지정할 때 업데이트된 DNS 레코드를 요청함으로써 발생하는 지연은 피할 수 있습니다. (예를 들어, DNS 장애 조치가 엔드포인트 중 하나에서 장애를 탐지하고 응답하는 경우 이러한 지연이 발생할 수 있습니다)
□-□-□	AWS 엔드포인트에 있는 디바이스의 중단이나 예정된 유지 관리에 대비한 리던던시를 제공할 수 있도록 두 개 이상의 VPN 터널이 구성되어 있습니다.
□-□-□	AWS Direct Connect를 사용하고 있으며 디바이스를 사용할 수 없는 경우 리던던시를 제공하도록 두 개의 Direct Connect 연결이 항상 구성되어 있습니다. 특정 위치를 사용할 수 없는 경우에 대비해 연결은 서로 다른 Direct Connect 위치에 프로비저닝되어 있습니다. 또한 여러 가상 인터페이스가 여러 Direct Connect 연결 및 위치에 걸쳐 구성되도록 가상 프라이빗 게이트웨이에 대한 연결을 구성했습니다.
□-□-□	Windows 인스턴스를 사용하고 있으며, 최신 PV 드라이버를 사용하고 있는지 확인합니다. PV 드라이버는 드라이버 성능 최적화에 도움이 되며 런타임 문제 및 보안 위험을 최소화합니다. 또한 Windows 인스턴스에서 EC2Config 에이전트의 최신 버전을 실행되고 있는지 확인합니다.
□-□-□	장애 시 point-in-time 복구를 보장하기 위해 Amazon Elastic Block Store(EBS) 볼륨의 스냅샷을 저장합니다.
□-□-□	운영 체제 및 애플리케이션/데이터베이스 데이터에 적절히 개별적인 Amazon EBS 볼륨을 사용합니다.
□-□-□	모든 Linux 인스턴스에 최신 커널, 소프트웨어 및 드라이버 패치를 적용합니다.

예-아니오-해당 없음	성능 효율성
□-□-□	AWS 호스팅 애플리케이션 구성 요소에 대하여 실행 전에 성능 테스트를 포함한 전체 테스트를 수행합니다. 또한 사용한 로드 테스트를 수행하여 올바른 EC2 인스턴스 크기, IOPS 수 및 RDS DB 인스턴스 크기 등을 사용하고 있는지 확인합니다.
□-□-□	서비스 한도에 대한 사용량 점검을 실행하고 AWS 서비스에 걸친 현재 사용량이 서비스 한도의 80% 미만인지 확인합니다. 자세히 알아보기
□-□-□	콘텐츠 전송/분산 네트워크(CDN)를 사용하여 애플리케이션(Amazon CloudFront) 캐싱을 활용하고 컨텐츠 전송을 최적화하며 가장 가까운 엣지 로케이션으로 컨텐츠가 자동 배포되도록 합니다.
□-□-□	Amazon CloudFront가 수신하는 일부 동적 HTTP 요청 헤더(사용자-에이전트, 날짜 등)는 캐시 적중률을 저하시키고 오리진에 부하를 가중하여 성능에 영향을 미칠 수 있음을 이해하고 있습니다. 자세히 알아보기
□-□-□	EC2 인스턴스의 최대 처리량이 연결된 EBS 볼륨의 최대 처리량 합계보다 큰지 확인합니다. 또한 EBS에 최적화된 인스턴스를 PIOPS EBS 볼륨에 사용하여 볼륨의 기대 성능을 실현합니다.
□-□-□	솔루션 설계에서 인프라 병목이 없으며 데이터베이스 또는 애플리케이션 설계에 스트레스 지점이 없는지 확인합니다.
□-□-□	애플리케이션 리소스에 대한 모니터링을 구현하고 Amazon CloudWatch 또는 타사 파트너

예-아니오-해당 없음	성능 효율성
	도구를 사용하여 성능 침해에 기반한 경보를 구성합니다.
□-□-□	애플리케이션에 연결된 보안 그룹에 대량의 규칙이 사용되지 않도록 하는 설계를 사용합니다. 보안 그룹에 많은 수의 규칙이 있으면 성능이 저하될 수 있습니다.

예-아니오-해당 없음	비용 최적화
□-□-□	인프라 이벤트는 일부 용량의 과도한 프로비저닝을 유발할 수 있으며 불필요한 비용을 피하기 위해 이벤트 후에 이러한 프로비저닝을 정리해야 한다는 점을 이해하고 있습니다.
□-□-□	EC2 인스턴스 크기, RDS DB 인스턴스 크기, 캐싱 클러스터 노드의 크기 및 수, Redshift 클러스터 노드의 크기와 수 및 EBS 볼륨 크기를 비롯하여 모든 인프라 구성 요소에 올바른 크기를 사용합니다.
□-□-□	필요하면 스팟 인스턴스를 사용합니다. 스팟 인스턴스는 유연한 시작 및 종료 시간을 가지는 워크로드에 이상적입니다. 스팟 인스턴스의 일반적인 사용 사례로는 배치 프로세싱, 보고서 생성 및 고성능 컴퓨팅 워크로드가 있습니다.
□-□-□	예측 가능한 애플리케이션 용량 최소 요구 사항이 파악되어 있으며 예약 인스턴스의 장점을 활용합니다. . 예약 인스턴스를 사용하면 온디맨드 인스턴스 요금보다 대폭 할인된 시간당 요금이 적용되는 Amazon EC2 컴퓨팅 용량을 예약할 수 잇습니다.

Notes

[1] https://aws.amazon.com/answers/account-management/aws-tagging-strategies/

[2] https://aws.amazon.com/blogs/aws/resource-groups-and-tagging/

[3] https://aws.amazon.com/sqs/

[4] http://docs.aws.amazon.com/general/latest/gr/rande.html

[5] https://aws.amazon.com/emr/

[6] https://aws.amazon.com/rds/

[7] https://aws.amazon.com/ecs/

[8] https://aws.amazon.com/sns/

[9] https://aws.amazon.com/blogs/compute/using-aws-lambda-with-auto-scaling-lifecycle-hooks/

[10] http://docs.aws.amazon.com/lambda/latest/dg/welcome.html

[11] https://aws.amazon.com/blogs/aws/new-auto-recovery-for-amazon-ec2/

[12] https://aws.amazon.com/answers/configuration-management/aws-infrastructure-configuration-management/

[13] https://d0.awsstatic.com/whitepapers/Big_Data_Analytics_Options_on_AWS%20.pdf

[14] http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-latency

[15] https://aws.amazon.com/elasticache/

[16] https://aws.amazon.com/cloudfront/

[17] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts-on-demand-reserved-instances.html

[18] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances.html

[19] https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html

[20] https://aws.amazon.com/about-aws/whats-new/2014/07/31/aws-trusted-advisor-security-and-service-limits-checks-now-free/

[21] http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_automated_manual.html

[22] http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html

[23] http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/publishingMetrics.html

[24] https://aws.amazon.com/blogs/aws/new-whitepaper-use-aws-for-disaster-recovery/

[25] http://media.amazonwebservices.com/AWS_Operational_Checklists.pdf

[26] http://d0.awsstatic.com/whitepapers/architecture/AWS_Well-Architected_Framework.pdf

[27] https://aws.amazon.com/premiumsupport/iem/