Amazon Neptune Analytics에 AWS Well-Architected 프레임워크 적용

AWS 권장 가이드

AWS 권장 가이드: Amazon Neptune Analytics에 AWS Well-Architected 프레임워크 적용

Table of Contents

Amazon Neptune Analytics에 AWS Well-Architected 프레임워크 적용

Michael Havey, Amazon Web Services(AWS)

2024년 12월(문서 기록)

Amazon Neptune을 사용하여 Amazon Web Services(AWS)에서 그래프 기반 솔루션을 빌드할 수 있습니다. Neptune에는 대량의 그래프 데이터를 빠르게 분석하여 인사이트를 얻고 추세를 찾을 수 있는 메모리 최적화 그래프 분석 엔진인 Neptune Analytics가 포함되어 있습니다. 기존 Neptune 데이터베이스 클러스터의 데이터에 대한 분석을 수행하거나 외부 데이터 세트의 데이터를 로드하고 분석할 수 있습니다. 이 가이드는 Neptune Analytics 배포를 계획할 때 AWS Well-Architected Framework 원칙을 적용하기 위한 규범적 지침을 제공합니다. Amazon Neptune용 AWS Well-Architected Framework를 적용하면 Neptune 데이터베이스에 대해 동일한 주제를 다룹니다.

AWS Well-Architected Framework를 사용하면 다양한 애플리케이션 및 워크로드를 위한 안전하고 성능이 뛰어나며 복원력이 뛰어나고 효율적인 인프라를 구축할 수 있습니다. 또한 키텍처를 평가하고 확장 가능한 설계를 구현할 수 있는 일관된 접근 방식을 제공합니다.

AWS Well-Architected 프레임워크는 다음 6가지 원칙을 중심으로 구축되었습니다.

- 운영 우수성
- 보안
- 신뢰성
- 성능 효율성
- 비용 최적화
- 지속 가능성

이 가이드는 Well-Architected Framework 설계 원칙 및 모범 사례의 정보와에 Neptune Analytics를 배포할 때 염두에 두어야 할 고려 사항을 제공합니다 AWS.

대상 독자

이 가이드는 그래프를 사용하는 솔루션을 설계하고 구현하는 데이터 엔지니어, 솔루션 아키텍트 및 데이터 분석가를 대상으로 합니다 AWS.

목표

이 가이드는 사용자와 조직이 다음을 수행하는 데 도움이 될 수 있습니다.

- 지원되는 배포 옵션 중에서 선택합니다.
- 복원력과 보안을 개선하는 데 도움이 되는 AWS Well-Architected 설계 패턴을 따르세요.
- 최적의 성능과 비용 절감을 위해 쿼리를 설계합니다.
- 프로덕션 환경에서 Neptune Analytics 그래프를 관리할 때 운영 효율성을 높이는 방법을 알아봅니다.

운영 우수성 요소

AWS Well-Architected Framework의 운영 우수성 원칙은 시스템을 실행 및 모니터링하고 프로세스와 절차를 지속적으로 개선하는 데 중점을 둡니다. 여기에는 효과적인 개발 및 워크로드 실행을 지원하고, 작업에 대한 인사이트를 얻으며, 지원 프로세스 및 절차를 지속적으로 개선하여 비즈니스 가치를 전달할 수 있는 능력이 포함됩니다. 사람의 개입 없이 대부분의 문제를 감지하고 해결하는 자가 복구 워크로드를 통해 운영 복잡성을 줄일 수 있습니다. 이 섹션에 설명된 모범 사례를 따르면이 목표를 달성할 수 있으며, Amazon Neptune Analytics 지표, APIs 및 메커니즘을 사용하여 워크로드가 예상 동작과 다를 때 적절하게 대응할 수 있습니다.

운영 우수성 원칙에 대한 이 논의는 다음 핵심 영역에 중점을 둡니다.

- 코드형 인프라(IaC)
- 변경 관리
- 복원력 전략
- 인시던트 관리
- 규정 준수를 위한 감사 보고
- 로깅 및 모니터링

IaC 접근 방식을 사용하여 배포 자동화

IaC를 사용하여 Neptune에서 배포를 자동화하는 모범 사례는 다음과 같습니다.

- IaC를 적용하여 Neptune Analytics 그래프 및 관련 리소스를 배포합니다. 일관된 환경 구성을 위해에서 제공하는 Neptune Analytics에 대한 지원을 사용하여 그래프와 프라이빗 엔드포인트AWS CloudFormation를 프로비저닝합니다.
- CloudFormation을 사용하여 Amazon SageMaker AI에서 Neptune 노트북 인스턴스를 프로비저닝합니다. 노트북을 사용하여 Neptune Analytics 그래프에서 데이터를 쿼리하고 시각화할 수 있습니다.
- Neptune 데이터베이스 클러스터, 스냅샷 또는 Amazon Simple Storage Service(Amazon S3)에 준비된 데이터 파일과 같은 기존 소스에서 Neptune 분석 그래프를 생성할 때 대량 가져오기 작업을 모니터링합니다.
- 그래프 크기 조정, 그래프 삭제 및 스냅샷 생성, 스냅샷에서 그래프 복원, 그래프 재설정 및 재로드와 같은 Neptune Analytics 운영 절차를 자동화합니다. AWS Command Line Interface (AWS CLI) 또는 SDK를 통해 사용할 수 있는 Neptune Analytics API를 사용합니다.

- 그래프의 필수 가동 시간을 평가합니다. 분석은 종종 일시적입니다. 그래프는 알고리즘을 실행해야 하는 시간 동안에만 필요합니다. 이 경우 AWS CLI 또는 SDKs를 사용하여 더 이상 필요하지 않을 때 그래프를 스냅샷 처리하고 삭제합니다. 그런 다음 필요한 경우 나중에 스냅샷에서 복원할 수 있습니다.
- 연결 문자열을 클라이언트 외부에 저장합니다. 연결 문자열을 AWS Secrets Manager, Amazon DynamoDB 또는 동적으로 변경할 수 있는 위치에 저장할 수 있습니다.
- 태그를 사용하여 Neptune Analytics 리소스에 메타데이터를 추가하고 태그를 기반으로 사용량을 추적합니다. 태그는 리소스를 구성하는 데 도움이 됩니다. 예를 들어 특정 환경 또는 애플리케이션의 리소스에 공통 태그를 적용할 수 있습니다. 태그를 사용하여 리소스 사용량에 대한 결제를 분석할 수도 있습니다. 자세한 내용은 AWS 결제 사용 설명서의 AWS 비용 할당 태그를 사용하여 비용 구성 및 추적을 참조하세요. 또한 AWS Identity and Access Management (IAM) 정책의 조건을 사용하여 해당 리소스에 사용되는 태그를 기반으로 리소스에 대한 액세스를 AWS 제어할 수 있습니다. 전역 `aws:ResourceTag/tag-key` 조건 키를 사용하면 됩니다. 자세한 내용은 IAM 사용 설명서의 AWS 리소스에 대한 액세스 제어를 참조하세요.

운영을 고려한 설계

Neptune Analytics 그래프 운영 방식을 개선하기 위한 접근 방식을 채택합니다.

- 개발, 테스트 및 프로덕션 용도로 별도의 Neptune Analytics 그래프를 유지 관리합니다. 이러한 그래프에는 데이터 세트, 사용자 및 운영 제어가 다를 수 있습니다.
- 다양한 용도에 맞게 별도의 Neptune Analytics 그래프를 유지 관리합니다. 예를 들어 두 분석 사용자 그룹이 서로 다른 타임라인, 모델, 성능 및 가용성 SLAs, 사용 패턴을 가진 별도의 그래프를 필요로 하는 경우 각 그룹에 대해 별도의 그래프를 유지합니다.
- Neptune Analytics 유지 관리 업데이트를 위해 사용자 및 운영 직원을 준비합니다.

되돌릴 수 있는 소규모 변경 자주 적용

다음 권장 사항은 복잡성을 최소화하고 워크로드 중단 가능성을 줄이기 위해 수행할 수 있는 작고 되돌릴 수 있는 변경 사항에 중점을 둡니다.

- IaC 템플릿 및 스크립트를 GitHub 또는 GitLab과 같은 소스 제어 서비스에 저장합니다.

> **Important**
>
> 소스 제어에 AWS 자격 증명을 저장하지 마십시오.

- AWS CodeDeploy 또는와 같은 지속적 통합 및 지속적 전달(CI/CD) 서비스를 사용하려면 IaC 배포가 필요합니다AWS CodeBuild. 프로덕션 그래프로 승격하기 전에 비프로덕션 Neptune Analytics 환경에서 코드를 컴파일, 테스트 및 배포합니다.

실행 가능한 인사이트를 위한 관찰성 구현

워크로드 동작, 성능, 신뢰성, 비용 및 상태를 포괄적으로 이해합니다. 다음 권장 사항은 Neptune Analytics에서 이러한 수준의 이해를 얻는 데 도움이 됩니다.

- Neptune Analytics에 대한 Amazon CloudWatch 지표를 모니터링합니다. 이러한 지표에서 그래프의 크기(노드, 엣지 및 벡터 수와 총 바이트 크기), CPU 사용률, 쿼리 요청 및 오류율을 확인할 수 있습니다.
- `NumQueuedRequestsPerSec`, , , `NumOpenCypherRequestsPerSec` `GraphStorageUsagePercent` `GraphSizeBytes`, 등의 주요 지표`CPUUtilization`와 애플리케이션 로그에 있는 Neptune 클라이언트 응답에 대한 CloudWatch 대시보드 및 경보를 생성합니다.
- 그래프 크기, 요청 속도 또는 CPU 사용률이 임계값을 초과하는 경우와 같이 Neptune Analytics 그래프의 상태를 모니터링하도록 알림을 설정합니다. 예를 들어 `GraphStorageUsagePercent`가 크게 성장하려는 그래프에서 90%로 상승한 경우 메모리 최적화 Neptune 용량 단위(m-NCU) 용량을 늘릴지 여부를 결정합니다. 현재 m-NCU가 128인 경우 256으로 늘리면 스토리지가 약 45% 줄어듭니다. `NumQueuedRequestsPerSec`가 0보다 큰 경우가 많으면 더 많은 컴퓨팅 용량을 제공하기 위해 m-NCU 용량을 늘리는 것이 좋습니다. 또는 클라이언트 측 동시성을 줄일 수 있습니다.

모든 운영 장애로부터 학습

자가 복구 인프라는 드문 문제가 발생하거나 응답이 원하는 만큼 효과적이지 않을 때 반복으로 발전하는 장기적인 노력을 보여줍니다. 다음 사례를 채택하면 다음과 같은 목표에 집중합니다.

- 모든 장애로부터 학습하여 개선을 주도합니다.
- 조직과 여러 팀에서 학습한 내용을 공유합니다. 조직 내 여러 팀이 Neptune을 사용하는 경우 공통 채팅룸 또는 사용자 그룹을 생성하여 학습 내용과 모범 사례를 공유합니다.

로깅 기능을 사용하여 무단 또는 비정상적인 활동 모니터링

로깅을 사용하여 비정상적인 성능 및 활동 패턴을 관찰합니다. 다음 모범 사례를 고려하세요.

- Neptune Analytics는를 사용하여 컨트롤 플레인 작업의 로깅을 지원합니다 AWS CloudTrail. 자세한 내용은를 사용하여 Neptune Analytics API 호출 로깅 AWS CloudTrail을 참조하세요. 이 기능을 통해 Neptune Analytics 리소스의 생성, 업데이트 및 삭제를 추적할 수 있습니다. 강력한 모니터링 및 경고를 위해 CloudTrail 이벤트를 Amazon CloudWatch Logs와 통합할 수도 있습니다. Neptune Analytics 서비스 활동에 대한 분석을 개선하고에 대한 활동의 변경 사항을 식별하기 위해 Amazon Athena를 사용하여 CloudTrail 로그를 쿼리 AWS 계정할 수 있습니다. 예를 들어 쿼리를 사용하여 트렌드를 식별하고 소스 IP 주소나 사용자 등의 속성별로 활동을 추가로 격리할 수 있습니다.
- CloudTrail을 사용하여 쿼리 실행과 같은 Neptune Analytics 데이터 영역 활동의 로깅을 활성화할 수도 있습니다. 실행 중인 쿼리, 빈도 및 소스를 볼 수 있습니다. 기본적으로 CloudTrail은 데이터 이벤트를 로깅하지 않습니다. 데이터 이벤트에는 추가 요금이 적용됩니다. 자세한 내용은 AWS CloudTrail 요금을 참조하십시오.
- 컨트롤 플레인 또는 데이터 플레인에서 Neptune Analytics에 대한 애플리케이션 호출을 로깅할 수도 있습니다. 예를 들어를 사용하여 쿼리AWS SDK for Python (Boto3)를 수행하는 경우 디버그 수준 로깅을 활성화하여 콘솔 또는 파일에 대한 쿼리 추적을 가져올 수 있습니다. 이는 개발 중에 유용합니다. 또한 애플리케이션에서 예외를 포착하고 기록하는 것이 좋습니다.

보안 요소

클라우드 보안이 가장 높은 우선 순위입니다 AWS. AWS 고객은 보안에 가장 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다. 보안은 AWS와 사용자의 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

- 클라우드 보안 - AWS 서비스 에서 실행되는 인프라를 보호할 AWS 책임이 있습니다 AWS 클라우드.는 안전하게 사용할 수 있는 서비스 AWS 도 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램의 일환으로 AWS 보안의 효과를 정기적으로 테스트하고 확인합니다. Neptune에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 AWS 규정 준수 프로그램 제공 범위 내 서비스를 참조하세요.
- 클라우드의 보안 - 사용자의 책임은 AWS 서비스 사용하는에 따라 결정됩니다. 또한 귀하는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 FAQs. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

AWS Well-Architected Framework의 보안 원칙은 Neptune Analytics를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 Neptune Analytics를 구성하는 방법을 설명합니다. 또한 Neptune Analytics 리소스를 모니터링하고 보호하는 데 도움이 AWS 서비스 되는 다른를 사용하는 방법을 알아봅니다. 보안 원칙에는 다음과 같은 주요 중점 영역이 포함됩니다.

- 데이터 보안
- 네트워크 보안
- 인증 및 권한 부여

데이터 보안 구현

데이터 유출 및 위반은 고객을 위험에 빠뜨리고 회사에 상당한 부정적인 영향을 미칠 수 있습니다. 다음 모범 사례는 우발적이고 악의적인 노출로부터 고객 데이터를 보호하는 데 도움이 됩니다.

- 그래프 이름, 태그, IAM 역할 및 기타 메타데이터에는 기밀 또는 민감한 정보가 포함되어서는 안 됩니다. 해당 데이터가 결제 또는 진단 로그에 표시될 수 있기 때문입니다.
- Neptune에 데이터로 저장된 외부 서버에 대한 URIs 또는 링크에는 요청을 검증하기 위한 자격 증명 정보가 포함되어서는 안 됩니다.

- Neptune 분석 그래프는 저장 시 암호화됩니다. 선택한 기본 키 또는 AWS Key Management Service (AWS KMS) 키를 사용하여 그래프를 암호화할 수 있습니다. 대량 가져오기 중에 Amazon S3로 내보내는 스냅샷과 데이터를 암호화할 수도 있습니다. 가져오기가 완료되면 암호화를 제거할 수 있습니다.
- openCypher 언어를 사용하는 경우 적절한 입력 검증 및 파라미터화 기술을 연습하여 SQL 주입 및 기타 형태의 공격을 방지합니다. 사용자가 제공한 입력과 문자열 연결을 사용하는 쿼리를 구성하지 마세요. 파라미터화된 쿼리 또는 준비된 문을 사용하여 입력 파라미터를 그래프 데이터베이스에 안전하게 전달합니다. 자세한 내용은 Neptune 설명서의 openCypher 파라미터화된 쿼리 예제를 참조하세요.

네트워크 보안

퍼블릭 연결을 위해 Neptune Analytics 그래프를 활성화하여 Virtual Private Cloud(VPC) 외부에서 연결할 수 있습니다. 이 연결은 기본적으로 비활성화되어 있습니다. 그래프에는 IAM 인증이 필요합니다. 호출자는 자격 증명을 얻고 그래프를 사용할 권한이 있어야 합니다. 예를 들어 openCypher 쿼리를 실행하려면 호출자에게 특정 그래프에 대한 읽기, 쓰기 또는 삭제 권한이 있어야 합니다.

그래프에 대한 프라이빗 엔드포인트를 생성하여 VPC 내에서 그래프에 액세스할 수도 있습니다. 엔드포인트를 생성할 때 VPC, 서브넷 및 보안 그룹을 지정하여 그래프 호출에 대한 액세스를 제한합니다.

전송 중 데이터를 보호하기 위해 Neptune Analytics는 HTTPS를 통해 그래프에 SSL 연결을 적용합니다. 자세한 내용은 Neptune Analytics 설명서의 Neptune Analytics의 데이터 보호를 참조하세요.

인증 및 권한 부여 구현

Neptune Analytics 그래프를 호출하려면 IAM 인증이 필요합니다. 호출자는 자격 증명을 얻고 그래프에서 작업을 수행할 수 있는 충분한 권한을 보유해야 합니다. API 작업 및 필요한 권한에 대한 설명은 Neptune Analytics API 설명서를 참조하세요. 조건 확인을 적용하여 태그별로 액세스를 제한할 수 있습니다.

IAM 인증은 AWS 서명 버전 4(SigV4) 프로토콜을 사용합니다. 애플리케이션의 사용을 간소화하려면 AWS SDK를 사용하는 것이 좋습니다. 예를 들어 Python에서 SigV4를 추상화하는 Neptune 그래프용 Boto3 클라이언트를 사용합니다.

그래프에 데이터를 로드할 때 배치 로드는 호출자의 IAM 자격 증명을 사용합니다. Neptune Analytics가 Amazon S3 파일에서 그래프로 데이터를 로드하는 역할을 수임할 수 있도록 신뢰 관계가 설정된 Amazon S3에서 데이터를 다운로드할 수 있는 권한이 호출자에게 있어야 합니다.

대량 가져오기는 그래프 생성 중(인프라 팀) 또는 기존 빈 그래프에서(가져오기 작업을 시작할 권한이 있는 데이터 엔지니어링 팀) 수행할 수 있습니다. 두 경우 모두 Neptune Analytics는 호출자가 입력으로 제공하는 IAM 역할을 수임합니다. 이 역할은 입력 데이터가 스테이징되는 Amazon S3 폴더의 콘텐츠를 읽고 나열할 수 있는 권한을 부여합니다.

신뢰성 요소

AWS Well-Architected Framework 신뢰성 원칙은 워크로드가 의도한 기능을 예상대로 올바르고 일관되게 수행할 수 있는 능력을 포함합니다. 여기에는 전체 수명 주기 동안 워크로드를 운영하고 테스트하는 기능이 포함됩니다.

신뢰할 수 있는 워크로드는 소프트웨어와 인프라에 대한 사전 설계 결정에서 시작됩니다. 아키텍처 선택은 모든 Well-Architected 원칙의 워크로드 동작에 영향을 미칩니다. 신뢰성을 위해이 단원에서 설명한 대로 따라야 하는 특정 패턴이 있습니다.

신뢰성 원칙은 다음 주요 영역에 중점을 둡니다.

- 서비스 할당량 및 배포 패턴을 포함한 워크로드 아키텍처
- 변경 관리
- 장애 관리

Neptune 서비스 할당량 이해

AWS 계정에는 각에 대한 기본 할당량(이전에는 제한이라고 함)이 있습니다 AWS 서비스. 다르게 표시되지 않는 한 리전별로 각 할당량이 적용됩니다. 모든 할당량이 아닌 일부 할당량에 대해 증가를 요청할 수 있습니다.

Neptune Analytics의 할당량을 찾으려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스를 선택한 다음 Amazon Neptune Analytics를 선택합니다. 그래프 및 스냅샷 수, 그래프에 대해 프로비저닝된 최대 메모리, API 요청 속도에 대한 할당량에 주의하십시오.

최대 프로비저닝 메모리가 데이터 세트에 충분하지 않은 경우 의도한 분석 사용에 필요한 노드 및 엣지 유형을 평가합니다. 허용되는 프로비저닝된 용량 내에서 분석이 가능하도록 데이터의 하위 집합을 로드합니다. 많은 분석 워크로드, 특히 그래프 알고리즘을 실행하는 워크로드에는 전체 트랜잭션 그래프 대신 제한된 속성 집합이 있는 토폴로지만 필요합니다. (트랜잭션 워크로드와 분석 워크로드의 차이점에 대한 설명은 성능 효율성 원칙 섹션을 참조하세요.)

최대 그래프 수가 의도한 용도에 충분하지 않은 경우:

- 비슷한 용도의 그래프를 결합하는 것이 좋습니다.
- 지정된 시간에 실행해야 하는 그래프 수를 평가합니다. 임시 분석 사용 사례가 있는 경우 그래프가 더 이상 필요하지 않을 때 그래프를 스냅샷 처리하고 삭제합니다. 이렇게 하면 할당량에 대한 그래프 수가 줄어듭니다.

- 그래프를 다른 로 프로비저닝하는 것이 좋습니다 AWS 계정.

Neptune 배포 패턴 이해

Neptune Analytics 그래프를 배포할 때 다음 결정 사항을 이해합니다.

- 시드: Amazon S3, 기존 Neptune 데이터베이스 클러스터 또는 기존 Neptune 데이터베이스 스냅샷의 데이터를 사용하여 생성 시 빈 그래프를 생성할지 아니면 여기에 데이터를 로드할지 결정합니다.

  권장 사항: 소스가 Neptune 클러스터 또는 스냅샷인 경우 그래프 생성 시 해당 데이터를 로드해야 합니다. 소스가 Amazon S3인 경우 데이터를 로드하는 작업이 중요하고 인프라 프로비저닝 활동으로 가장 잘 수행되는 경우 생성 시 데이터를 로드합니다. 데이터를 데이터 엔지니어링 또는 애플리케이션 활동으로 로드하려면 나중에 빈 그래프를 생성하고 Amazon S3에서 데이터를 로드합니다.
- 용량: 데이터 크기와 예상 애플리케이션 사용량을 고려하여 그래프에 필요한 프로비저닝된 용량을 추정합니다.

  권장 사항: 생성 시 그래프 크기를 제한할 최대 프로비저닝 메모리를 지정합니다. 이 설정은 필수입니다. 필요한 경우 나중에 용량을 변경할 수 있습니다.
- 가용성 및 내결함성: 가용성을 위해 복제본이 필요한지 여부를 결정합니다. 복제본은 그래프 실패 시 복구를 위한 웜 스탠바이 역할을 합니다. 복제본이 있는 그래프는 복제본이 없는 그래프보다 빠르게 복구됩니다. 또한 그래프가 얼마나 오래 필요한지, 임시 분석 전용인지 여부, 필요한 경우 언제 제거되는지도 고려합니다.

  권장 사항: 그래프를 생성하기 전에 그래프를 사용할 수 없는 시간 및 제거할 수 있는 시기와 같은 가용성 요구 사항을 결정합니다.
- 네트워킹 및 보안: 퍼블릭 연결, 프라이빗 연결 또는 둘 다 필요한지 여부와 데이터를 암호화할지 여부를 결정합니다.

  권장 사항: 그래프를 생성하기 전에 퍼블릭 연결이 허용되는지 여부, 그래프 클라이언트 애플리케이션이 배포될 위치 등 조직의 요구 사항을 이해합니다.
- 백업 및 복구: 스냅샷을 생성해야 하는지 여부와 생성해야 하는 경우 언제 또는 어떤 조건에서 생성해야 하는지 결정합니다. 조직에 재해 복구(DR) 요구 사항이 있는지 고려합니다.

  권장 사항: 스냅샷 생성은 수동 활동입니다. 그래프를 생성하기 전에 스냅샷을 생성할 시기를 결정하고 DR 요구 사항을 고려합니다.

Neptune 클러스터 관리 및 확장

Neptune 분석 그래프는 메모리에 최적화된 단일 인스턴스로 구성됩니다. 인스턴스의 용량(m-NCU)은 생성 시 설정됩니다. 관리 작업을 통해 프로비저닝된 용량을 늘려 인스턴스를 수직으로 확장할 수 있으며 프로비저닝된 용량도 줄일 수 있습니다. 복제본은 수동 장애 조치 대상이므로 그래프의 크기를 늘리지 않습니다. 이와 관련하여 그래프 복제본은 애플리케이션의 읽기 작업을 처리할 수 있는 Neptune 클러스터의 활성 인스턴스인 Neptune 데이터베이스 읽기 전용 복제본과 다릅니다.

복제본에는 비용이 발생합니다. 복제본은 그래프의 m-NCU 요금으로 가격이 책정됩니다. 예를 들어 그래프가 128m-NCU에 프로비저닝되고 단일 복제본이 있는 경우 비용은 복제본이 없는 동등한 그래프의 두 배입니다.

분석에는 확장해야 하는 두 가지 주요 이유가 있습니다.

- 분석 쿼리 및 알고리즘에 더 많은 메모리와 CPU를 제공하기 위해 개별 쿼리는 비용이 많이 들기 때문에 실행할 그래프 알고리즘은 본질적으로 복잡하며 입력 시 더 많은 리소스가 필요하거나 동시 요청 속도가 높습니다. 이러한 쿼리에 out-of-memory 오류가 발생하는 경우 스케일 업이 합리적인 해결 방법입니다.
- 계획한 것보다 더 큰 그래프 크기를 지원합니다. 예를 들어 현재 프로비저닝된 용량이 60GB의 소스 데이터를 지원하는 128m-NCU이고 추가 40GB의 소스 데이터가 필요한 경우 256m-NCU로 늘려야 합니다.

, , NumQueuedRequestsPerSec, NumOpenCypherRequestsPerSecGraphStorageUsagePercentGraphSizeBytes,와 같은 Neptune Analytics의 CloudWatch 지표를 모니터링CPUUtilization하여 조정이 필요한지 확인합니다. 콘솔 AWS CLI또는 SDKs. (예제 및 모범 사례는 운영 우수성 원칙 섹션을 참조하세요.)

백업 및 장애 조치 이벤트 관리

복제본을 사용하여 실패 시 그래프를 계속 사용할 수 있도록 합니다. 그래프는 로그 기반 지속성을 사용하여의 가용 영역 간에 변경 사항을 커밋합니다 AWS 리전. 복제본은 웜 스탠바이 역할을 하며이 데이터에 액세스할 수 있습니다. 오류가 있는 경우 그래프는 복제본에 대한 작업을 재개합니다. 애플리케이션은 동일한 엔드포인트를 계속 사용하여 그래프에 연결합니다. 실패 중 진행 중인 요청은 서비스 사용 불가 예외와 함께 오류를 생성합니다. 애플리케이션 코드에서 백오프 패턴과 함께 재시도를 사용하여 오류를 포착하고 짧은 간격 후에 다시 시도하는 것이 좋습니다. 장애 조치 중에 이루어진 새 요청은 대기열에 추가되며 지연 시간이 길어질 수 있습니다.

복제본이 구성되지 않고 그래프가 실패하면 Neptune Analytics는 내구성 있는 스토리지에서 복구되지만 Neptune이 리소스를 다시 초기화해야 하므로 복구 시간이 더 오래 걸립니다.

그래프의 스냅샷을 생성합니다. (Neptune Analytics는 자동 스냅샷을 생성하지 않습니다.) 생성 후 그래프를 정기적으로 수정하는 경우 스냅샷을 자주 생성하여 현재 상태를 캡처합니다. 이전 시점으로 복원할 필요가 없는 경우 이전 스냅샷을 삭제합니다.

스냅샷을 다른 계정 및 여러 계정과 공유할 수 있습니다 AWS 리전. DR 요구 사항이 있는 경우 스냅샷에서 다른 리전으로 그래프를 복원하는 것이 Recovery Time Objective(RTO) 및 Recovery Point Objective(RPO) 요구 사항을 충족하는지 고려합니다.

성능 효율성 요소

AWS Well-Architected Framework의 성능 효율성 원칙은 데이터를 수집하거나 쿼리하는 동안 성능을 최적화하는 방법에 중점을 둡니다. 성능 최적화는 다음과 같은 점진적이고 지속적인 프로세스입니다.

- 비즈니스 요구 사항 확인
- 워크로드 성능 측정
- 성능이 낮은 구성 요소 식별
- 비즈니스 요구 사항에 맞게 구성 요소 튜닝

성능 효율성 원칙은 사용할 올바른 그래프 데이터 모델 및 쿼리 언어를 식별하는 데 도움이 되는 사용 사례별 지침을 제공합니다. 또한 Neptune Analytics로 데이터를 수집하고 사용할 때 따라야 할 모범 사례도 포함되어 있습니다.

성능 효율성 원칙은 다음 핵심 영역에 중점을 둡니다.

- 그래프 모델링
- 쿼리 최적화
- 그래프 오른쪽 크기 조정
- 쓰기 테이블 최적화

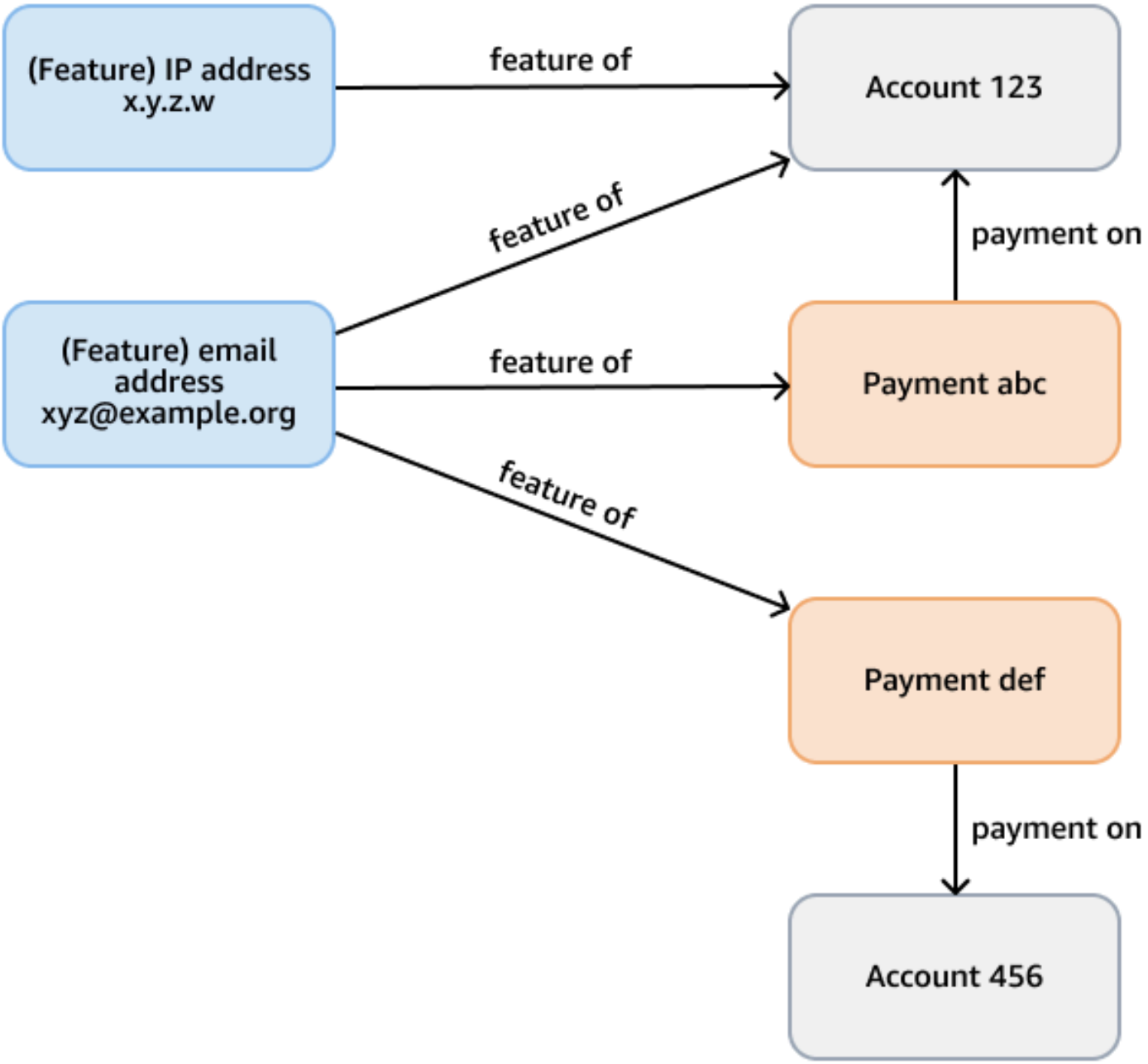
분석을 위한 그래프 모델링 이해

Amazon Neptune에 AWS Well-Architected 프레임워크 적용 가이드에서는 성능 효율성을 위한 그래프 모델링에 대해 설명합니다. 성능에 영향을 미치는 모델링 결정에는 필요한 노드 및 엣지 선택, IDs, 레이블 및 속성, 엣지 방향, 레이블이 일반인지 특정인지 여부, 일반적으로 쿼리 엔진이 그래프를 탐색하여 일반적인 쿼리를 처리할 수 있는 효율성이 포함됩니다.

이러한 고려 사항은 Neptune Analytics에도 적용되지만 트랜잭션 및 분석 사용 패턴을 구분하는 것이 중요합니다. Neptune 데이터베이스와 같은 트랜잭션 데이터베이스의 쿼리에 효율적인 그래프 모델은 분석을 위해 재구성해야 할 수 있습니다.

예를 들어 신용 카드 결제에서 사기 패턴을 확인하는 것이 목적인 Neptune 데이터베이스의 사기 그래프를 생각해 보세요. 이 그래프에는 계정과 결제 모두의 계정, 결제 및 기능(예: 이메일 주소, IP 주소, 전화번호)을 나타내는 노드가 있을 수 있습니다. 이 연결된 그래프는 지정된 결제에서 시작하고 여러

홉을 통해 관련 기능 및 계정을 찾는 가변 길이 경로를 통과하는 등의 쿼리를 지원합니다. 다음 그림은 이러한 그래프를 보여줍니다.



기능으로 연결된 계정의 커뮤니티를 찾는 등 분석 요구 사항이 더 구체적일 수 있습니다. 이를 위해 WCC(약하게 연결된 구성 요소) 알고리즘을 사용할 수 있습니다. 이전 예제의 모델에 대해 실행하려면 여러 유형의 노드와 엣지를 통과해야 하므로 비효율적입니다. 다음 다이어그램의 모델이 더 효율적입니다. 계정 자체 또는 계정의 결제가 기능을 공유하는 경우 `account` 노드를 `shares feature` 엣지와 연결합니다. 예를 들어 `Account 123`에는 이메일 기능이 `xyz@example.org`있으며 결제()에 동일한 이메일을 `Account 456` 사용합니다`Payment def`.

WCC의 계산 복잡성은 이며O(|E|logD), 여기서 |E|는 그래프의 엣지 수이고 D는 노드를 연결하는 직경(가장 긴 경로의 길이)입니다. 트랜잭션 모델은 가상 노드와 엣지를 생략하기 때문에 엣지 수와 직경을 모두 최적화하고 WCC 알고리즘의 복잡성을 줄입니다.

Neptune Analytics를 사용하는 경우 필요한 알고리즘 및 분석 쿼리에서 다시 작업합니다. 필요한 경우 모델을 재구성하여 이러한 쿼리를 최적화합니다. 그래프에 데이터를 로드하기 전에 모델을 재구성하거나 그래프의 기존 데이터를 수정하는 쿼리를 작성할 수 있습니다.

쿼리 최적화

다음 권장 사항에 따라 Neptune Analytics 쿼리를 최적화합니다.

- 파라미터화된 쿼리와 기본적으로 활성화된 쿼리 계획 캐시를 사용합니다. 계획 캐시를 사용하면 엔진이 나중에 사용할 수 있도록 쿼리를 준비합니다. 단, 쿼리가 100밀리초 이내에 완료되므로 후속 호출에 소요되는 시간이 절약됩니다.
- 느린 쿼리의 경우 설명 계획을 실행하여 병목 현상을 파악하고 그에 따라 개선합니다.
- 벡터 유사성 검색을 사용하는 경우 작은 임베딩이 정확한 유사성 결과를 산출하는지 확인합니다. 더 작은 임베딩을 더 효율적으로 생성, 저장 및 검색할 수 있습니다.
- Neptune Analytics에서 openCypher를 사용하기 위한 문서화된 모범 사례를 따릅니다. 예를 들어 UNWIND 절에서 평면화된 맵을 사용하고 가능한 경우 엣지 레이블을 지정합니다.
- 그래프 알고리즘을 사용할 때는 알고리즘의 입력 및 출력, 컴퓨팅 복잡성, 작동 방식을 광범위하게 이해해야 합니다.
  - 그래프 알고리즘을 호출하기 전에 MATCH 절을 사용하여 입력 노드 세트를 최소화합니다. 예를 들어 노드가 폭 우선 검색(BFS)을 수행하도록 제한하려면 Neptune Analytics 설명서에 제공된 예제를 따르세요.
  - 가능하면 노드 및 엣지 레이블을 기준으로 필터링합니다. 예를 들어 BFS에는 특정 노드 레이블(vertexLabel) 또는 특정 엣지 레이블()로의 순회를 필터링하는 입력 파라미터가 있습니다edgeLabels.
  - 와 같은 경계 파라미터를 사용하여 결과를 제한maxDepth합니다.

- `concurrency` 파라미터로 실험합니다. 사용 가능한 모든 알고리즘 스레드를 사용하여 처리를 병렬화하는 값 0으로 시도합니다. 파라미터를 1로 설정하여 단일 스레드 실행과 비교합니다. 알고리즘은 단일 스레드에서, 특히 병렬 처리로 실행 시간이 크게 단축되지 않고 오버헤드가 발생할 수 있는 얕은 너비 우선 검색과 같은 작은 입력에서 더 빠르게 완료될 수 있습니다.
- 유사한 유형의 알고리즘 중에서 선택합니다. 예를 들어 Bellman-Ford와 delta-stepping은 모두 단일 소스 최단 경로 알고리즘입니다. 자체 데이터 세트로 테스트할 때는 두 알고리즘을 모두 시도하고 결과를 비교합니다. Delta-stepping은 계산 복잡성이 낮기 때문에 Bellman-Ford보다 빠른 경우가 많습니다. 그러나 성능은 데이터 세트 및 입력 파라미터, 특히 `delta` 파라미터에 따라 달라집니다.

쓰기 최적화

Neptune Analytics에서 쓰기 작업을 최적화하려면 다음 사례를 따르세요.

- 그래프에 데이터를 로드하는 가장 효율적인 방법을 찾습니다. Amazon S3의 데이터에서 로드할 때 데이터가 50GB보다 큰 경우 대량 가져오기를 사용합니다. 더 작은 데이터의 경우 배치 로드를 사용합니다. 배치 로드를 실행할 때 out-of-memory 오류가 발생하는 경우 m-NCU 값을 늘리거나 로드를 여러 요청으로 분할하는 것이 좋습니다. 이를 수행하는 한 가지 방법은 S3 버킷의 여러 접두사로 파일을 분할하는 것입니다. 이 경우 각 접두사에 대해 배치 로드를 개별적으로 호출합니다.
- 대량 가져오기 또는 배치 로더를 사용하여 초기 그래프 데이터 세트를 채웁니다. 트랜잭션 openCypher 생성, 업데이트 및 삭제 작업은 소규모 변경에만 사용합니다.
- 대량 가져오기 또는 동시성이 1(단일 스레드)인 배치 로더를 사용하여 그래프에 임베딩을 수집합니다. 다음 방법 중 하나를 사용하여 임베딩을 미리 로드해 보세요.
- 벡터 유사성 검색 알고리즘에서 정확한 유사성 검색에 필요한 벡터 임베딩의 차원을 평가합니다. 가능하면 더 작은 차원을 사용합니다. 따라서 임베딩의 로드 속도가 빨라집니다.
- 필요한 경우 변형 알고리즘을 사용하여 알고리즘 결과를 기억합니다. 예를 들어 도 변형 중심 알고리즘은 각 입력 노드의 정도를 찾아 해당 값을 노드의 속성으로 씁니다. 이러한 노드를 둘러싼 연결이 이후에 변경되지 않으면 속성에 올바른 결과가 포함됩니다. 알고리즘을 다시 실행할 필요가 없습니다.
- 다시 시작해야 하는 경우 그래프 재설정 관리 작업을 사용하여 모든 노드, 엣지 및 임베딩을 지웁니다. 그래프가 큰 경우 openCypher 쿼리를 사용하여 모든 노드, 엣지 및 임베딩을 삭제할 수 없습니다. 대규모 데이터 세트에 대한 단일 드롭 쿼리는 시간 초과될 수 있습니다. 크기가 증가함에 따라 데이터 세트를 제거하는 데 시간이 더 오래 걸리고 트랜잭션 크기가 증가합니다. 반대로 그래프 재설정을 완료하는 데 걸리는 시간은 거의 일정하며, 작업은 스냅샷을 실행하기 전에 스냅샷을 생성하는 옵션을 제공합니다.

적절한 크기의 그래프

전체 성능은 Neptune 분석 그래프의 프로비저닝된 용량에 따라 달라집니다. 용량은 메모리 최적화 Neptune 용량 단위(m-NCUs)라는 단위로 측정됩니다. 그래프의 크기가 그래프 크기와 쿼리를 지원할 수 있을 만큼 충분한지 확인합니다. 용량 증가가 개별 쿼리의 성능을 반드시 개선하는 것은 아닙니다.

가능하면 Amazon S3 또는 기존 Neptune 클러스터 또는 스냅샷과 같은 기존 소스에서 데이터를 가져와 그래프를 생성합니다. 최소 및 최대 용량에 경계를 둘 수 있습니다. 기존 그래프에서 프로비저닝된 용량을 변경할 수도 있습니다.

`NumQueuedRequestsPerSec`, , , `NumOpenCypherRequestsPerSec` `GraphStorageUsagePercent` `GraphSizeBytes`, 등의 CloudWatch 지표를 모니터링`CPUUtilization`하여 그래프의 크기가 적절한지 평가합니다. 그래프 크기와 로드를 지원하는 데 더 많은 용량이 필요한지 확인합니다. 이러한 지표 중 일부를 해석하는 방법에 대한 자세한 내용은 운영 우수성 원칙 섹션을 참조하세요.

비용 최적화 요소

AWS Well-Architected Framework의 비용 최적화 원칙은 불필요한 비용을 방지하는 데 중점을 둡니다. 다음 권장 사항은 Neptune Analytics의 비용 최적화 설계 원칙과 아키텍처 모범 사례를 충족하는 데 도움이 될 수 있습니다.

비용 최적화 원칙은 다음 주요 영역에 중점을 둡니다.

- 시간 경과에 따른 지출 이해 및 자금 할당 제어
- 올바른 유형 및 수량의 리소스 선택
- 과도한 지출 없이 비즈니스 요구 사항을 충족하도록 규모 조정

필요한 사용 패턴 및 서비스 이해

Neptune Analytics를 채택하기 전에 사용 사례가 그래프 분석에 적합한지 평가합니다.

- 그래프 데이터베이스: 데이터 모델에 식별 가능한 그래프 구조가 있고 쿼리가 관계를 탐색하고 여러 홉을 통과해야 하는 경우 Neptune과 같은 그래프 데이터베이스가 워크로드에 적합합니다. 그래프 데이터베이스는 다음 패턴에 적합하지 않습니다.
  - 주로 단일 홉 쿼리입니다. 이 사용 사례에서는 데이터가 객체의 속성으로 더 잘 표현될 수 있는지 고려합니다.
  - 속성으로 저장된 JSON 또는 이진 대형 객체(blob) 데이터입니다.

- 그래프 분석: Neptune Analytics는 메모리에서 대량의 그래프 데이터를 빠르게 분석하여 인사이트를 얻고 추세를 찾을 수 있는 그래프 분석 데이터베이스 엔진입니다. Neptune 데이터베이스와 Neptune Analytics 그래프 모두에 그래프 데이터를 저장하고 쿼리할 수 있습니다. Neptune 데이터베이스는 확장 가능한 온라인 트랜잭션 처리(OLTP) 요구 사항에 가장 적합합니다. Neptune Analytics는 임시 분석 워크로드에 가장 적합합니다. 트랜잭션 지향 Neptune 데이터베이스의 데이터를 Neptune 분석 그래프로 로드하여 두 데이터를 조합하여 해당 데이터에 대한 분석을 실행할 수 있습니다. 분석이 완료되면 Neptune 분석 그래프를 제거할 수 있습니다. 자세한 비교는 Neptune Analytics 설명서의 Neptune Analytics를 사용하는 시기와 Neptune 데이터베이스를 사용하는 시기를 참조하세요.

비용을 고려하여 Neptune Analytics 그래프를 채우는 가장 좋은 방법을 결정합니다.

- S3 버킷에 준비된 그래프 데이터를 대량으로 가져옵니다. 데이터가 이전에 Neptune 데이터베이스에 대량 로드하기 위해 준비된 경우 또는 대량 가져오기에 필요한 CSV 또는 기타 지원되는 형식으로 분석할 데이터가 이미 있거나 쉽게 생성할 수 있는 경우이 옵션을 사용하는 것이 좋습니다. 그래프 생성 절차의 일부로 대량 가져오기를 실행할 수 있습니다. 최소 및 최대 용량에 경계를 둘 수 있습니다. 이전에 생성한 빈 그래프에서 가져오기를 실행하고 실행 중에 가져오기 작업을 모니터링할 수도 있습니다.
- 빈 그래프를 생성한 다음 배치 로드를 사용하여 openCypher 쿼리를 통해 그래프를 채울 수 있습니다. 이 옵션은 로드할 데이터가 Amazon S3에서 스테이징되고 50GB 미만인 경우에 적합합니다.
- Neptune 데이터베이스 클러스터의 데이터에서 그래프를 채울 수 있습니다(Neptune 데이터베이스 버전 1.3.0 이상에서 지원됨). 이 패턴의 의도는 현재 그래프 데이터베이스에 있는 데이터에 대한 분석을 실행하는 것입니다. 데이터베이스가 처음에 대량 로드를 통해 채워졌더라도 이후 크게 변경되었을 수 있습니다. 데이터베이스에서 가져오기 위해 Neptune Analytics는 데이터베이스를 복제하고 복제본에서 S3 버킷으로 데이터를 내보냅니다. 이 절차에는 특히 복제본을 실행하는 데 드는 Neptune 데이터베이스 비용과 내보낸 데이터를 저장하고 사용하는 데 드는 Amazon S3 비용이 발생합니다. 내보내기가 완료되면 복제본이 제거됩니다. Amazon S3에서 내보낸 데이터를 삭제할 수 있습니다.
- Neptune 데이터베이스 클러스터의 스냅샷에서 그래프를 채울 수 있습니다. 이는 소스가 데이터베이스 스냅샷이라는 점을 제외하면 이전 옵션과 유사합니다. 스냅샷에서 가져오기 위해 Neptune Analytics는 먼저 스냅샷을 새 데이터베이스 클러스터로 복원한 다음 데이터를 S3 버킷으로 내보냅니다. 이 절차에는 특히 복원된 클러스터를 실행하는 데 드는 Neptune 데이터베이스 비용과 내보낸 데이터를 저장하고 사용하는 데 드는 Amazon S3 비용이 발생합니다.
- 또한 openCypher 쿼리를 수행하여 그래프에서 원자성, 일관성, 격리, 내구성(ACID) 준수 트랜잭션을 사용하여 데이터를 생성, 업데이트 또는 삭제할 수 있습니다. 이 접근 방식은 그래프를 시드하는 방법이 아니라 소규모 업데이트를 수행하는 방법으로 사용하는 것이 좋습니다.

분석에 필요한 데이터가 Amazon S3에 이미 준비된 경우 대량 가져오기 또는 배치 로드를 사용하는 것이 좋습니다. 이는 Neptune 데이터베이스 클러스터 또는 스냅샷에서 그래프를 채우는 것보다 비용 효율적입니다.

비용에 주의를 기울이는 리소스 선택

Neptune Analytics 요금은 메모리 최적화 Neptune 용량 단위(m-NCU)라는 단위를 사용합니다. 지정된 m-NCU로 그래프를 실행하는 데는 고정 시간당 비용이 부과됩니다. 그래프에는 장애 조치를 위한 복제본이 있을 수 있으며 이러한 복제본에는 시간당 m-NCU 비용도 발생합니다.

용량을 추정하고, 비용을 제한하고, 성능을 기준으로 비용을 모니터링하려면 다음 모범 사례를 따르는 것이 좋습니다.

- 가능하면 Amazon S3에 스테이징된 데이터, 기존 Neptune 클러스터 또는 스냅샷 등 기존 소스에서 데이터를 가져와 그래프를 생성합니다. 이렇게 하면 Neptune Analytics가 그래프를 시드하는 작업을 많이 수행하고 제한 최대 용량을 지정할 수 있으므로 노력이 절약됩니다.
- 기존 그래프에서 프로비저닝된 용량을 변경할 수 있습니다.
- 그래프가 더 이상 필요하지 않은 경우 스냅샷을 생성하고 그래프를 삭제할 수 있습니다. 다시 사용해야 하는 경우 스냅샷에서 그래프를 복원할 수 있습니다.
- 그래프를 생성할 때 복제본 수를 선택할 수 있습니다. 분석 가용성 요구 사항에 따라 값을 설정합니다. 이 설정을 최소화하여 비용을 절감합니다. 최대값 2는 별도의 가용 영역에 있는 복제본 인스턴스 2개를 허용합니다. 최소값 0은 Neptune Analytics가 복제본을 실행하지 않음을 의미합니다. 하지만 복제본을 사용할 수 있으면 복구 속도가 빨라집니다. 그래프 장애 및 복구에 대한 설명은 신뢰성 원칙 섹션을 참조하세요.
- 를 사용하여 현재 및 과거 결제 기간의 Neptune Analytics 비용을 모니터링합니다AWS 결제 및 비용 관리.
- CloudWatch, 특히 `NumQueuedRequestsPerSec`, `CPUUtlization`, `NumOpenCypherRequestsPerSec`, `GraphSizeBytes`, `GraphStorageUsagePercent`에 대한 Neptune Analytics 지표를 모니터링하여 프로비저닝된 용량이 그래프에 적합한 크기인지 평가합니다. 더 작은 용량이 관찰된 요청 속도, CPU 사용량 및 그래프 크기를 수용할 수 있는지 확인합니다.
- 그래프에 프라이빗 엔드포인트가 필요한 경우 탄력적 IPs, Virtual Private Cloud(VPC) 엔드포인트, NAT 게이트웨이 또는 기타 VPC 관련 비용에 주의하십시오. 자세한 내용은 Amazon VPC 요금 및 Amazon EC2 요금을 참조하세요.
- 개발자와 분석가가 그래프를 쿼리하고 시각화하는 데 도움이 되는 클라이언트 인터페이스를 제공하기 위해 하나 이상의 Neptune 노트북 인스턴스를 실행할 수 있습니다(Neptune 워크벤치 요금 참조). 비용을 최소화하려면 사용자 간에 인스턴스를 공유하고 각 사용자에 대해 별도의 노트북 폴더를 생성합니다. 인스턴스를 사용하지 않을 때는 종료합니다. 종료를 자동화하는 방법은 블로그 게시물 리소스 태그를 사용하여 Amazon Neptune 환경 리소스의 중지 및 시작 자동화를 참조 AWS 하세요.

지속 가능성 요소

AWS Well-Architected Framework의 지속 가능성 원칙은 클라우드 워크로드 실행이 환경에 미치는 영향을 최소화하는 데 중점을 둡니다. 주요 주제에는 지속 가능성에 대한 공동 책임 모델, 영향 이해, 사용 극대화로 필요한 리소스를 최소화하고 다운스트림 영향을 줄이는 것이 포함됩니다.

지속 가능성 원칙에는 다음과 같은 주요 핵심 영역이 포함됩니다.

- 사용자의 영향
- 지속 가능성 목표
- 최대 사용량
- 새롭고 보다 효율적인 소프트웨어 제품 예측 및 채택
- 관리형 서비스 사용
- 다운스트림 영향 감소

이 가이드는 영향을 이해하는 데 중점을 둡니다. 기타 지속 가능성 설계 원칙에 대한 자세한 내용은 AWS Well-Architected 프레임워크를 참조하세요.

사용자의 선택 사항과 요구 사항은 환경에 영향을 미칩니다. 탄소 강도 AWS 리전 가 낮은를 선택할 수 있고 요구 사항이 가동 시간과 내구성만 극대화하는 대신 실제 워크로드 요구 사항을 반영하는 경우 워크로드의 지속 가능성이 증가합니다. 다음 섹션에서는 워크로드 설계 및 지속적 운영에 채택될 경우 환경에 긍정적인 영향을 미치는 모범 사례 및 고려 사항에 대해 설명합니다.

AWS 리전 선택 고려

일부는 Amazon 재생 에너지 프로젝트 AWS 리전 근처에 있거나 그리드의 탄소 강도가 다른 프로젝트보다 낮은 곳에 있습니다. 워크로드에 실행 가능할 수 있는 리전의 지속 가능성에 미치는 영향을 고려하고 목록을 Neptune Analytics를 사용할 수 있는 리전과 상호 참조하세요.

소비 최적화

다음을 연습하여 Neptune 분석 사용을 최소화합니다.

- 분석은 종종 일시적입니다. 그래프는 알고리즘을 실행하고 결과를 기록하는 시간에만 필요합니다. 이 경우 그래프가 더 이상 필요하지 않을 때 그래프를 스냅샷 처리하고 삭제합니다. 나중에 필요한 경우 스냅샷에서 복원할 수 있습니다.

- 워크로드가 일시적이고 분석을 실행할 시기를 유연하게 결정할 수 있는 경우 전력 소비의 day-to-day 추세를 고려하세요. 특정 시간 동안 전기 수요가 더 높습니다. 미국에 있는 경우 미국 에너지 정보 관리(EIA) 웹 사이트에서 일일 전기 소비에 대한 지표를 참조하세요. 가능하면 해당 리전의 사용량이 적은 기간에 워크로드를 실행합니다.
- 워크로드가 임시 워크로드는 아니지만 제한된 기간 동안만 사용할 수 있어야 하는 경우 그래프를 삭제하고 필요할 때 스냅샷에서 복원합니다. 가용성이 일정을 따르는 경우 예약된 시간에 그래프가 준비되도록 스크립트를 통해 복원 프로세스를 자동화합니다.
- 데이터가 읽기 전용이거나 마지막 스냅샷 이후 변경되지 않은 경우 삭제하기 전에 다시 스냅샷을 생성하지 마십시오.
- Neptune 노트북을 사용하지 않을 때는 중지합니다.
- `NumQueuedRequestsPerSec`, , `NumOpenCypherRequestsPerSec` `GraphStorageUsagePercent` `GraphSizeBytes`, 및와 같은 CloudWatch 지표를 모니터링`CPUUtilization`하여 그래프가 큰지 평가합니다. 더 작은 인스턴스 용량이 관찰된 요청 속도, CPU 사용량 및 그래프 크기를 수용할 수 있는지 확인합니다.

소프트웨어 개발 및 아키텍처 패턴 최적화

낭비를 방지하려면 모델 및 쿼리를 최적화하고 컴퓨팅 리소스를 공유하여 Neptune 인스턴스 및 클러스터에서 사용할 수 있는 모든 리소스를 사용합니다. 구체적인 모범 사례는 다음과 같습니다.

- 쿼리 및 그래프 알고리즘 호출을 최적화합니다. 파라미터화된 쿼리를 사용하고 기본적으로 활성화된 쿼리 계획 캐시를 사용합니다. 느린 쿼리의 경우 설명 계획을 실행하여 개선합니다. 벡터 유사성 검색을 사용하는 경우 더 작은 임베딩을 더 효율적으로 생성, 저장 및 검색할 수 있으므로 더 작은 임베딩이 정확한 유사성 결과를 제공하는지 확인합니다. 그래프 알고리즘을 호출하기 전에 `MATCH` 절을 사용하여 입력 노드 세트를 최소화합니다. 가능하면 노드 및 엣지 레이블을 기준으로 필터링합니다.
- 그래프에 데이터를 로드하는 가장 효율적인 방법을 찾습니다. Amazon S3의 데이터에서 로드하는 경우 데이터가 50GB보다 큰 경우 대량 가져오기를 사용합니다. 더 작은 데이터에 배치 로드를 사용합니다.
- 개발자에게 자체 인스턴스를 생성하는 대신 Neptune 노트북 인스턴스를 공유하도록 요청합니다. 단일 Jupyter 인스턴스에서 각 개발자에 대해 별도의 노트북 폴더를 생성합니다. 인스턴스를 사용하지 않을 때는 종료합니다.

리소스

참조

- AWS Well-Architected
- AWS Well-Architected Framework 설명서
- Amazon Neptune용 AWS Well-Architected 프레임워크 적용
- Neptune 분석 모범 사례

블로그 게시물 및 동영상

- Neptune 블로그 게시물(AWS 데이터베이스 블로그)
- 리소스 태그를 사용하여 Amazon Neptune 환경 리소스의 중지 및 시작 자동화(AWS 데이터베이스 블로그)
- Amazon Neptune 다과류(짧은 YouTube 동영상)

학습

- Getting Started with Amazon Neptune
- Amazon Neptune으로 빌드
- Data Modeling for Amazon Neptune
- Amazon Neptune 분석 워크숍

문서 기록

아래 표에 이 가이드의 주요 변경 사항이 설명되어 있습니다. 향후 업데이트에 대한 알림을 받으려면 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
최초 게시	—	2024년 12월 20일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.