Amazon EC2 상에 SQL 서버 배포 모범 사례

AWS 규범적 지침

AWS 규범적 지침: Amazon EC2 상에 SQL 서버 배포 모범 사례

Table of Contents

Amazon EC2 Microsoft SQL Server 배포의 모범 사례

Abhishek Soni 및 Sagar Patel, Amazon Web Services (AWS)

2023년 12월(문서 기록)

이 설명서의 목적은 Amazon Web Services(AWS) 클라우드에서 Amazon Elastic Compute Cloud(Amazon EC2)로 Microsoft SQL Server를 배포하거나 마이그레이션한 후 일관된 환경을 보장하는 것입니다. 이 설명서는 데이터베이스 및 서버 구성의 모범 사례를 제공하여 인프라를 최적화하고, 성능을 조정하고, 배포 또는 마이그레이션 후 예상치 못한 문제가 발생하지 않도록 합니다.

이 설명서는 Microsoft SQL Server를 온프레미스 환경에서 Amazon EC2로 마이그레이션할 계획이거나 Amazon EC2에서 새로운 SQL Server 배포를 최적화하려는 데이터베이스 아키텍트, 시스템 및 데이터베이스 책임자, 관리자를 위한 것입니다.

Amazon EC2는 AWS 클라우드에서 확장 가능한 컴퓨팅 용량을 제공합니다. Amazon EC2에서 SQL Server를 사용하는 것은 온프레미스에서 SQL Server를 실행하는 것과 비슷합니다. Amazon EC2를 사용하면 인프라와 데이터베이스 환경을 완벽하게 제어할 수 있습니다. AWS 클라우드의 규모, 성능, 탄력성을 활용할 수는 있지만 EC2 인스턴스, 스토리지 볼륨, 파일 시스템, 네트워킹, 보안을 포함한 모든 구성 요소를 구성하고 조정하는 것은 사용자의 책임입니다. 이 설명서는 구성을 최적화하고 AWS에서 SQL Server 성능을 극대화하는 데 도움이 되는 정보를 제공합니다. 서버 및 스토리지 설정과 모범 사례에 대해 자세히 설명합니다. 또한, 해당하는 경우 설정을 자동화하는 방법을 설명하고 데이터베이스 수준에서의 구성 변경에 대해서도 설명합니다.

> **Note**
>
> 또한 AWS는 온프레미스 SQL Server 데이터베이스를 SQL Server용 Amazon Relational Database Service(RDS)와 같은 관리형 서비스로 이동하는 옵션을 제공합니다. 마이그레이션 옵션에 대한 설명은 AWS 권장 지침 웹사이트의 관계형 데이터베이스를 위한 마이그레이션 전략을 참고하십시오.

컴퓨팅 및 스토리지 설정 구성

Amazon EC2에 SQL Server를 마이그레이션하거나 배포하기 전에 EC2 인스턴스 및 스토리지 설정을 구성하여 성능을 개선하고 비용을 절감할 수 있습니다. 다음 섹션에서는 최적화 팁과 모범 사례를 제공합니다.

주제

- Amazon EBS 최적 인스턴스 타입 사용
- 디스크 레이아웃 또는 파일 배포 최적화
- NTFS 할당 단위 크기를 64KB로 설정하십시오.
- tempdb를 인스턴스 스토어에 배치하십시오.
- CPU 코어 불일치 방지
- 디스크 성능 테스트
- 인스턴트 파일 초기화 활성화
- 메모리에서 페이지 잠그기
- TCP 오프로드 및 RSS 설정 비활성
- IOPS 및 처리량 요건 결정
- 스트라이핑을 사용하여 IOPS 및 처리량 제한을 우회할 수 있습니다.
- 바이러스 백신 소프트웨어에서 SQL Server 파일 제외

Amazon EBS 최적 인스턴스 타입 사용

SQL Server 데이터베이스가 I/O 집약적인 워크로드를 처리하는 경우, Amazon Elastic Block Store(Amazon EBS)에 최적화된 인스턴스를 프로비저닝하면 성능을 개선하는 데 도움이 됩니다.

Amazon EBS 최적화 인스턴스는 최적화된 구성 스택을 사용하며 Amazon EBS I/O를 위한 전용 용량을 추가로 제공합니다. 이 최적화는 Amazon EBS I/O와 인스턴스의 다른 트래픽 사이의 경합을 최소화하여 EBS 볼륨에 최상의 성능을 제공합니다.

디스크 레이아웃 또는 파일 배포 최적화

한 볼륨은 데이터 및 로그 파일에 사용하고, 다른 볼륨은 tempdb 워크로드에 사용하고, 콜드 HDD(sc1) 또는 처리량 최적화 HDD(st1) 볼륨은 백업에 사용합니다.

I/O 관련 문제가 발생하여 데이터 및 로그 파일의 워크로드를 분리하려는 경우, 다른 볼륨을 사용하는 것이 좋습니다. 워크로드로 인해 특정 데이터베이스를 분리해야 하는 경우, 각 데이터베이스에 전용 볼륨을 사용하는 것이 좋습니다.

일반적으로 tempdb는 가장 높은 I/O의 대상이므로 해당 워크로드가 분리되지 않으면 병목 현상이 발생할 수 있습니다. 이렇게 분리하면 tempdb를 사용자 데이터베이스의 데이터 및 로그 파일에서 분리하는 데도 도움이 됩니다. 비교적 저렴한 백업용 스토리지를 사용하여 비용을 최적화할 수 있습니다.

NTFS 할당 단위 크기를 64KB로 설정하십시오.

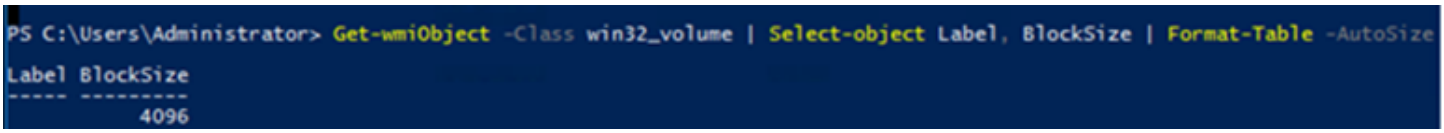
SQL Server에서 스토리지의 원자 단위는 1개 페이지이며 크기는 8KB입니다. 물리적으로 인접한 8개의 페이지가 익스텐트(크기 64KB)를 구성합니다. SQL Server는 익스텐트를 사용하여 데이터를 저장합니다. 따라서 SQL Server 시스템에서 SQL 데이터베이스 파일(tempdb 포함)을 호스팅하기 위한 NTFS 할당 단위 크기는 64KB여야 합니다.

PowerShell 또는 명령행을 사용하여 드라이브의 클러스터(NTFS 할당) 크기를 확인할 수 있습니다.

PowerShell 사용:

```
Get-wmiObject -Class win32_volume | Select-object Label, BlockSize | Format-Table -
AutoSize
```

다음 그림에 PowerShell의 출력 예가 나와 있습니다.



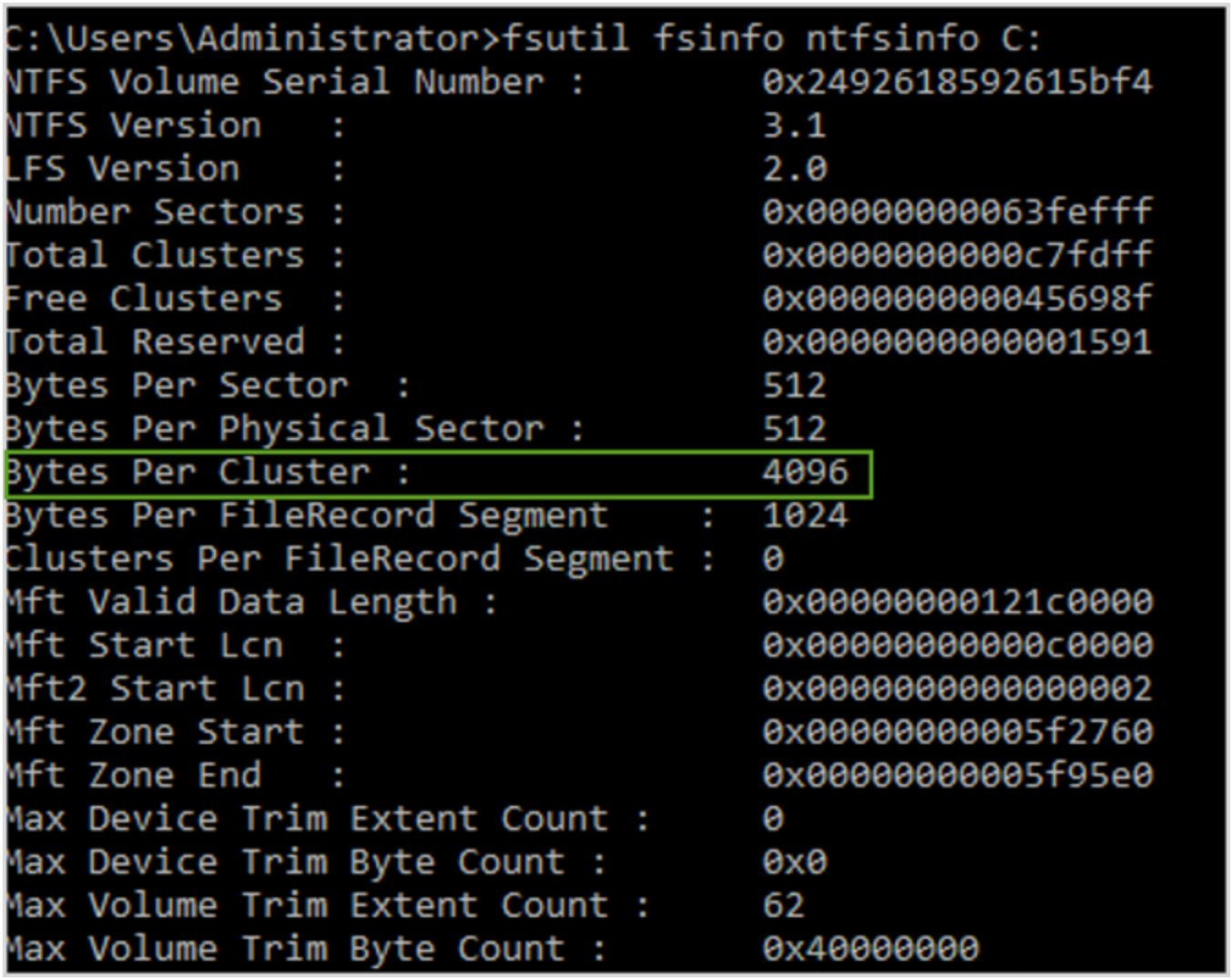
또는 다음을 사용하십시오.

```
$wmiQuery = "SELECT Name, Label, BlockSize FROM win32_volume WHERE FileSystem='NTFS'"
Get-wmiObject -Query $wmiQuery -ComputerName '.' | Sort-Object Name | Select-Object
 Name, Label, BlockSize
```

명령행 사용:

```
$ fsutil fsinfo ntfsinfo C:
```

다음 그림은 명령행의 출력 예를 보여줍니다. 클러스터당 바이트 값은 형식 크기를 바이트 단위로 표시합니다. 예 출력은 4096바이트입니다. SQL Server 데이터베이스 파일을 호스팅하는 드라이브의 경우, 이 값은 64KB여야 합니다.



Amazon EC2에서 SSD 스토리지를 사용할 때 SQL Server 성능이 블록 크기에 좌우되지 않는 경우도 있습니다. 자세한 설명은 블로그 게시물을 참조하십시오. AWS 고객은 64KB 블록 크기의 SQL Server 스토리지를 활용할 수 있습니까?

tempdb를 인스턴스 스토어에 배치하십시오.

Amazon EC2 인스턴스 스토어를 사용하는 경우, tempdb용 인스턴스 스토어 볼륨을 사용하십시오. 인스턴스 스토어는 인스턴스에 블록 수준의 임시(단기간) 스토리지를 제공합니다. 속도와 비용이라는 두 가지 이유로 인스턴스 스토어 볼륨에 tempdb를 배치하는 것이 좋습니다. tempdb는 일반적으로 가장 많이 사용되는 데이터베이스이므로 사용 가능한 가장 빠른 드라이브를 활용할 수 있습니다. tempdb를

인스턴스 스토어에 배치할 때 얻을 수 있는 또 다른 이점은 인스턴스 스토어에 대한 I/O 요금이 별도로 청구되지 않기 때문에 비용이 절감된다는 것입니다.

tempdb는 SQL Server를 다시 시작할 때마다 다시 생성되므로 인스턴스를 중지하거나 종료해도 데이터가 손실되지 않습니다. 하지만 임시 디스크가 시스템에 로컬로 연결되어 있기 때문에 다른 호스트에서 가상 시스템을 시작하면 인스턴스 스토어 볼륨이 손실되므로 신중하게 계획하십시오.

인스턴스 스토어 볼륨을 사용하는 경우:

- SQL Server 서비스가 시작되기 전에 볼륨을 초기화하십시오. 그렇지 않으면 SQL Server 시작 절차가 실패합니다.
- 인스턴스 스토어 볼륨에 대한 권한(전체 제어)을 SQL Server 시작 계정에 명시적으로 부여합니다.

tempdb를 인스턴스 스토어로 이동

tempdb를 인스턴스 스토어 볼륨으로 이동하려면:

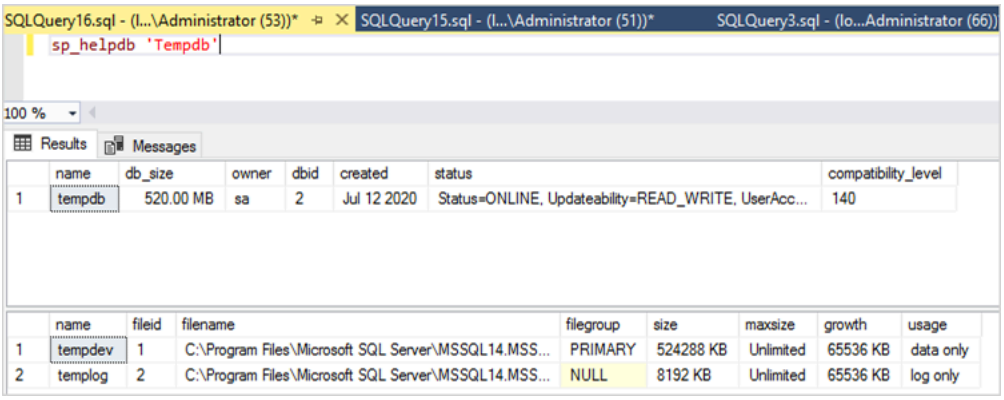
1. Windows에서 관리자 권한으로 `diskmgmt.msc`를 실행하여 디스크 관리 시스템 유틸리티를 엽니다.
2. 새 디스크를 초기화합니다.
3. 디스크를 마우스 오른쪽 버튼으로 클릭한 다음 새 단순 볼륨을 선택합니다.
4. 다음 설정을 사용하여 프롬프트를 완료하여 볼륨을 포맷합니다.

   - 파일 시스템: NTFS
   - 할당 단위 크기: 64K
   - 볼륨 레이블: tempdb

   자세한 설명은 Microsoft 웹사이트에서 디스크 관리 설명서를 참조하십시오.
5. SQL Server 인스턴스에 연결하고 다음 명령을 실행하여 tempdb 데이터베이스의 논리적 및 물리적 파일 이름을 기록합니다.

   ```
   $ sp_helpdb 'tempdb'
   ```

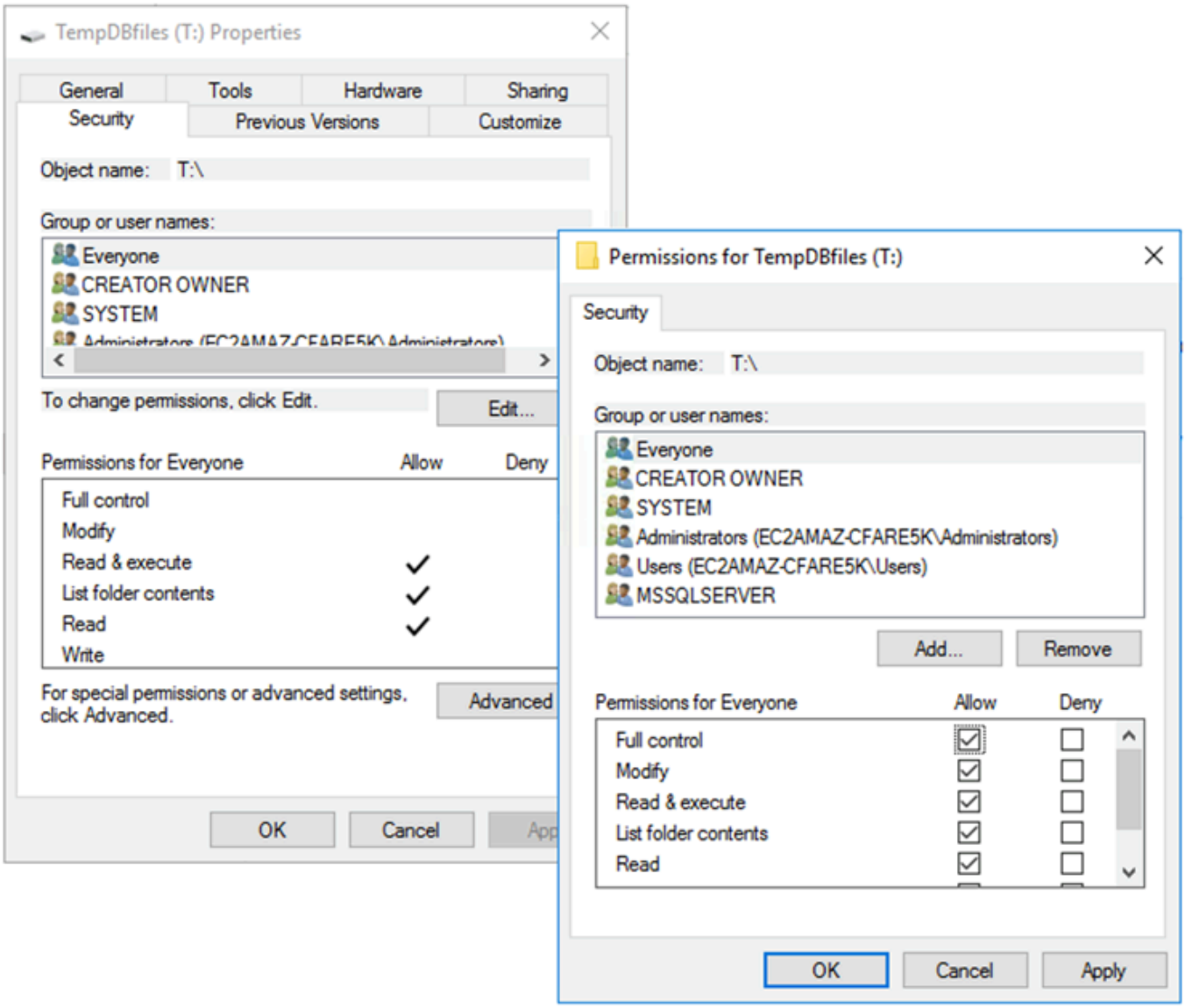
   다음 스크린샷은 명령과 그 출력을 보여줍니다.



6. tempdb 파일을 새 위치로 이동합니다. 모든 tempdb 데이터베이스 파일을 동일한 초기 크기로 설정해야 한다는 점을 기억하십시오. 다음 샘플 SQL 서버 스크립트는 tempdb 파일을 T 드라이브로 이동하고 데이터 파일을 같은 크기로 설정합니다.

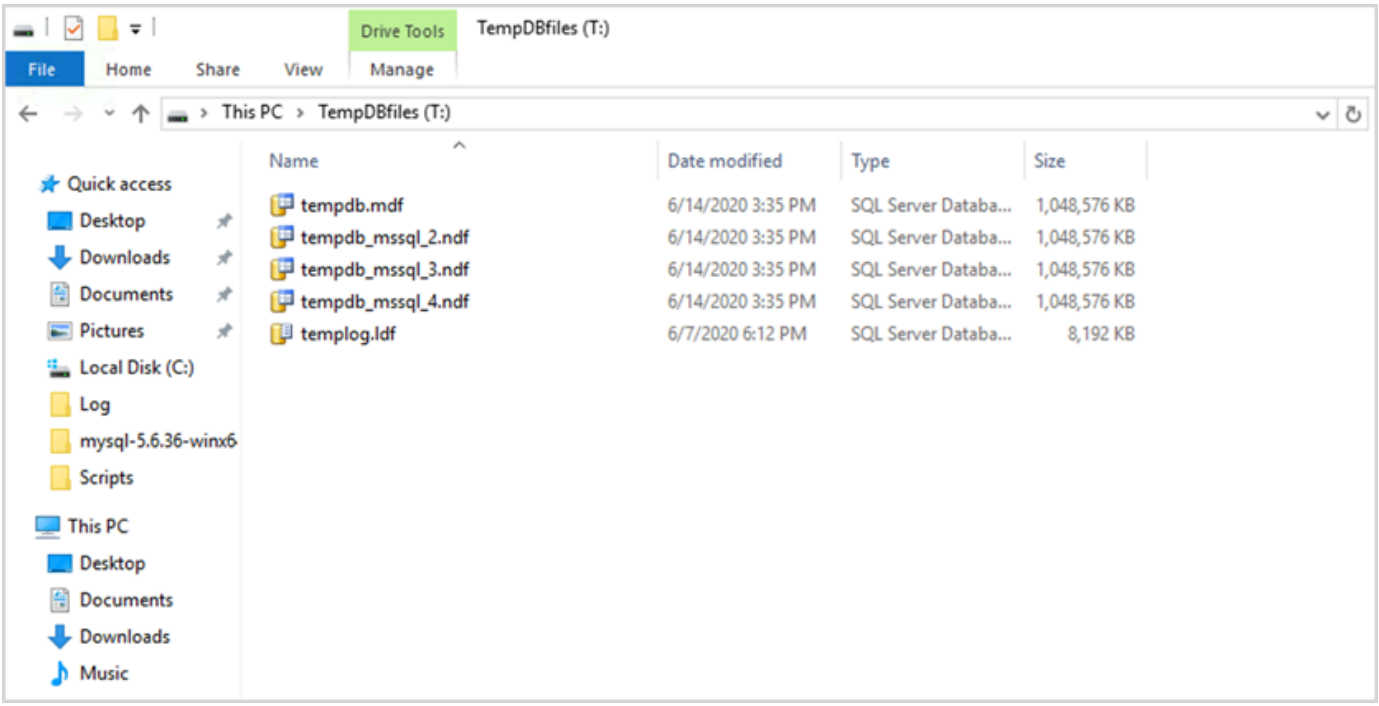
```
USE master
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = tempdev, FILENAME = 'T:\tempdb.mdf',SIZE
 = 524288KB)
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = temp2, FILENAME = 'T:
\tempdb_mssql_2.ndf',SIZE = 524288KB)
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = temp3, FILENAME = 'T:
\tempdb_mssql_3.ndf',SIZE = 524288KB)
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = temp4, FILENAME = 'T:
\tempdb_mssql_4.ndf',SIZE = 524288KB)
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = templog, FILENAME = 'T:\templog.ldf')
GO
```

7. SQL Server 시작 계정에 tempdb 데이터베이스의 새 위치에 대한 권한을 부여하면 다음 스크린샷과 같이 tempdb 파일을 만들 수 있습니다.



8. 새 위치의 tempdb를 사용하려면 SQL Server를 다시 시작하십시오.

   다음 스크린샷과 같이 새 위치에 작성된 tempdb 파일을 볼 수 있습니다.



9. 이전 위치에서 tempdb 파일을 삭제합니다.

인스턴스 재부팅 또는 시작/중지 시 SQL Server가 시작되기 전에 인스턴스 스토어 볼륨이 초기화되도록 하려면 다음 섹션의 단계를 따르십시오. 그렇지 않으면 tempdb가 초기화되지 않아 SQL Server 시작이 실패합니다.

인스턴스 스토어 초기화

데이터 스토어를 초기화하려면:

1. Windows 서비스 관리자(`services.msc`)를 열고 SQL Server 및 해당 종속 서비스(예: SQL Server 에이전트)를 수동으로 시작하도록 설정합니다. (인스턴스 스토어 볼륨이 준비되면 스크립트를 사용하여 시작합니다.)
2. PowerShell 스크립트를 생성하여 Amazon EC2 인스턴스에 사용자 데이터로 전달합니다. 이 스크립트는 다음을 수행합니다.
   - 임시 스토리지를 탐지하고 이를 위한 tempdb 드라이브(이 예에서는 T 드라이브)를 생성합니다.
   - EC2 인스턴스가 중지되었다가 다시 시작되는 경우, 임시 디스크를 새로 고칩니다.
   - SQL Server 시작 계정에 새로 초기화된 tempdb 볼륨에 대한 전체 제어 권한을 부여합니다. 이 예에서는 기본 인스턴스를 가정하므로 `NT SERVICE\MSSQLSERVER`를 사용합니다. 명명된 인스턴스의 경우, 이 값은 기본적으로 `NT SERVICE\MSSQL$<InstanceName>`입니다.

- 스크립트를 로컬 볼륨(예의 `c:\scripts`)에 저장하고 파일 이름(`InstanceStoreMapping.ps1`)을 할당합니다.
- Windows 작업 스케줄러를 사용하여 예약된 작업을 만듭니다. 이 작업은 시작 시 PowerShell 스크립트를 실행합니다.
- 이전 작업을 수행한 후 SQL Server 및 SQL Server 에이전트를 시작합니다.

다음 스크립트는 MS-SQL 가용성 그룹 워크숍의 두 번째 실습에서 가져온 것이며 일부 내용이 변경되었습니다. EC2 인스턴스를 시작할 때 스크립트를 사용자 데이터 필드에 복사하고 필요에 따라 맞춤합니다.

```
<powershell>
# Create pool and virtual disk for TempDB using the local NVMe, ReFS 64K, T: Drive
    $NVMe = Get-PhysicalDisk | ? { $_.CanPool -eq $True -and $_.FriendlyName -eq "NVMe
 Amazon EC2 NVMe"}
    New-StoragePool -FriendlyName TempDBPool -StorageSubsystemFriendlyName "Windows
 Storage*" -PhysicalDisks $NVMe
    New-VirtualDisk -StoragePoolFriendlyName TempDBPool -FriendlyName TempDBDisk -
ResiliencySettingName simple -ProvisioningType Fixed -UseMaximumSize
    Get-VirtualDisk -FriendlyName TempDBDisk | Get-Disk | Initialize-Disk -Passthru
 | New-Partition -DriveLetter T -UseMaximumSize | Format-Volume -FileSystem ReFS -
AllocationUnitSize 65536 -NewFileSystemLabel TempDBfiles -Confirm:$false
    # Script to handle NVMe refresh on start/stop instance
    $InstanceStoreMapping = {
    if (!(Get-Volume -DriveLetter T)) {
        #Create pool and virtual disk for TempDB using mirroring with NVMe
        $NVMe = Get-PhysicalDisk | ? { $_.CanPool -eq $True -and $_.FriendlyName -eq
 "NVMe Amazon EC2 NVMe"}
        New-StoragePool -FriendlyName TempDBPool -StorageSubsystemFriendlyName "Windows
 Storage*" -PhysicalDisks $NVMe
        New-VirtualDisk -StoragePoolFriendlyName TempDBPool -FriendlyName TempDBDisk -
ResiliencySettingName simple -ProvisioningType Fixed -UseMaximumSize
        Get-VirtualDisk -FriendlyName TempDBDisk | Get-Disk | Initialize-Disk -Passthru
 | New-Partition -DriveLetter T -UseMaximumSize | Format-Volume -FileSystem ReFS -
AllocationUnitSize 65536 -NewFileSystemLabel TempDBfiles -Confirm:$false
         #grant SQL Server Startup account full access to the new drive
        $item = gi -literalpath "T:\"
        $acl = $item.GetAccessControl()
        $permission="NT SERVICE\MSSQLSERVER","FullControl","Allow"
        $rule = New-Object System.Security.AccessControl.FileSystemAccessRule
 $permission
```

```
        $acl.SetAccessRule($rule)
        $item.SetAccessControl($acl)
        #Restart SQL so it can create tempdb on new drive
        Stop-Service SQLSERVERAGENT
        Stop-Service MSSQLSERVER
        Start-Service MSSQLSERVER
        Start-Service SQLSERVERAGENT
        }
    }
    New-Item -ItemType Directory -Path c:\Scripts
    $InstanceStoreMapping | set-content c:\Scripts\InstanceStoreMapping.ps1
# Create a scheduled task on startup to run script if required (if T: is lost)
    $action = New-ScheduledTaskAction -Execute 'Powershell.exe' -Argument 'c:\scripts
\InstanceStoreMapping.ps1'
    $trigger =  New-ScheduledTaskTrigger -AtStartup
    Register-ScheduledTask -Action $action -Trigger $trigger -TaskName "Rebuild
 TempDBPool" -Description "Rebuild TempDBPool if required" -RunLevel Highest -User
 System
</powershell>
```

버퍼 풀 확장 사용

버퍼 풀 확장을 사용할 계획이라면 임시 볼륨에 배치하는 것도 고려해 볼 수 있습니다. 하지만 구현하기 전에 철저하게 테스트하는 것이 좋습니다. 버퍼 풀 확장과 tempdb에 같은 볼륨을 사용하지 마십시오.

> **Note**
>
> 버퍼 풀 확장은 경우에 따라 유용할 수 있지만 RAM을 대체할 수는 없습니다. 사용을 결정하기 전에 Microsoft 웹사이트에 제공된 세부 정보를 참조하십시오.

CPU 코어 불일치 방지

라이선스가 지원하는 것보다 코어 수가 많은 서버를 선택하면 CPU 왜곡이 발생하여 CPU 성능이 낭비될 수 있습니다. 이는 논리적 코어와 실제 코어 사이의 매핑 때문입니다. SQL Server를 CAL(클라이언트 액세스 라이선스)과 함께 사용하면 일부 스케줄러는 `VISIBLE ONLINE`이 되고 나머지는 `VISIBLE OFFLINE`이 됩니다. 이로 인해 스케줄러 노드가 최적으로 활용되지 않아 NUMA(비균일 메모리 액세스) 토폴로지에서 성능 문제가 발생할 수 있습니다.

예컨대, `m5.24xlarge` 인스턴스에서 SQL Server를 실행하면 코어가 24개인 소켓 2개와 소켓당 논리 프로세서 48개가 검색되어 총 96개의 논리 프로세서가 생성됩니다. 48코어에 대한 라이선스만 보유한 경우, SQL Server 오류 로그에 다음과 비슷한 메시지가 표시됩니다.

2020-06-08 12:35:27.37 Server SQL Server가 SQL Server 라이선스를 기준으로 48개의 논리적 프로세서를 사용하여 소켓당 24개의 코어와 소켓당 48개의 논리 프로세서가 있는 소켓 2개, 총 96개의 논리 프로세서를 감지했습니다. 이는 정보를 제공하는 메시지임; 사용자 조치가 요구되지 않음.

총 코어 수와 SQL Server에서 사용 중인 코어 수 사이에 차이가 있는 경우, CPU 사용량 불균형을 확인하거나 라이선스가 지원하는 코어 수와 같은 서버 타입을 사용하십시오.

CPU 스큐: 예(`m5.24xlarge`)의 인스턴스 타입에 대해 SQL Server는 기본적으로 8개의 NUMA 노드를 생성합니다. 이러한 노드 중 4개(상위 노드 ID 0,1,2,3)에만 `VISIBLE ONLINE` 상태의 스케줄러가 있습니다. 나머지 스케줄은 모두 `VISIBLE OFFLINE`입니다. 스케줄러들 사이의 이러한 차이는 성능 저하로 이어질 수 있습니다.

스케줄러 정보 및 상태를 확인하려면 다음을 사용하십시오.

```
$ select * from sys.dm_os_schedulers
```

SQL Server 라이선스에서 지원하는 것보다 많은 수의 코어가 있는 서버 인스턴스를 사용하려면 Amazon EC2 설명서의 인스턴스를 위한 CPU 옵션 지정에 나와 있는 지침에 따라 코어 수를 맞춤하는 것이 좋습니다.

디스크 성능 테스트

DiskSPD와 같은 도구를 사용하여 디스크 성능을 확인하는 것이 좋습니다. 이 도구를 사용하면 SQL Server별 테스트를 실행하기 전에 디스크 속도를 추정할 수 있습니다. EBS 볼륨은 온프레미스 환경에서 기존 SAN과 다르게 작동하므로 디스크 성능을 평가하는 것이 매우 중요합니다. 적절한 성능 테스트가 이루어지지 않으면 마이그레이션 후 추정치 못한 성능 저하로 이어질 수 있습니다. DiskSPD를 사용하여 맞춤 테스트를 실행할 수도 있습니다.

인스턴트 파일 초기화 활성화

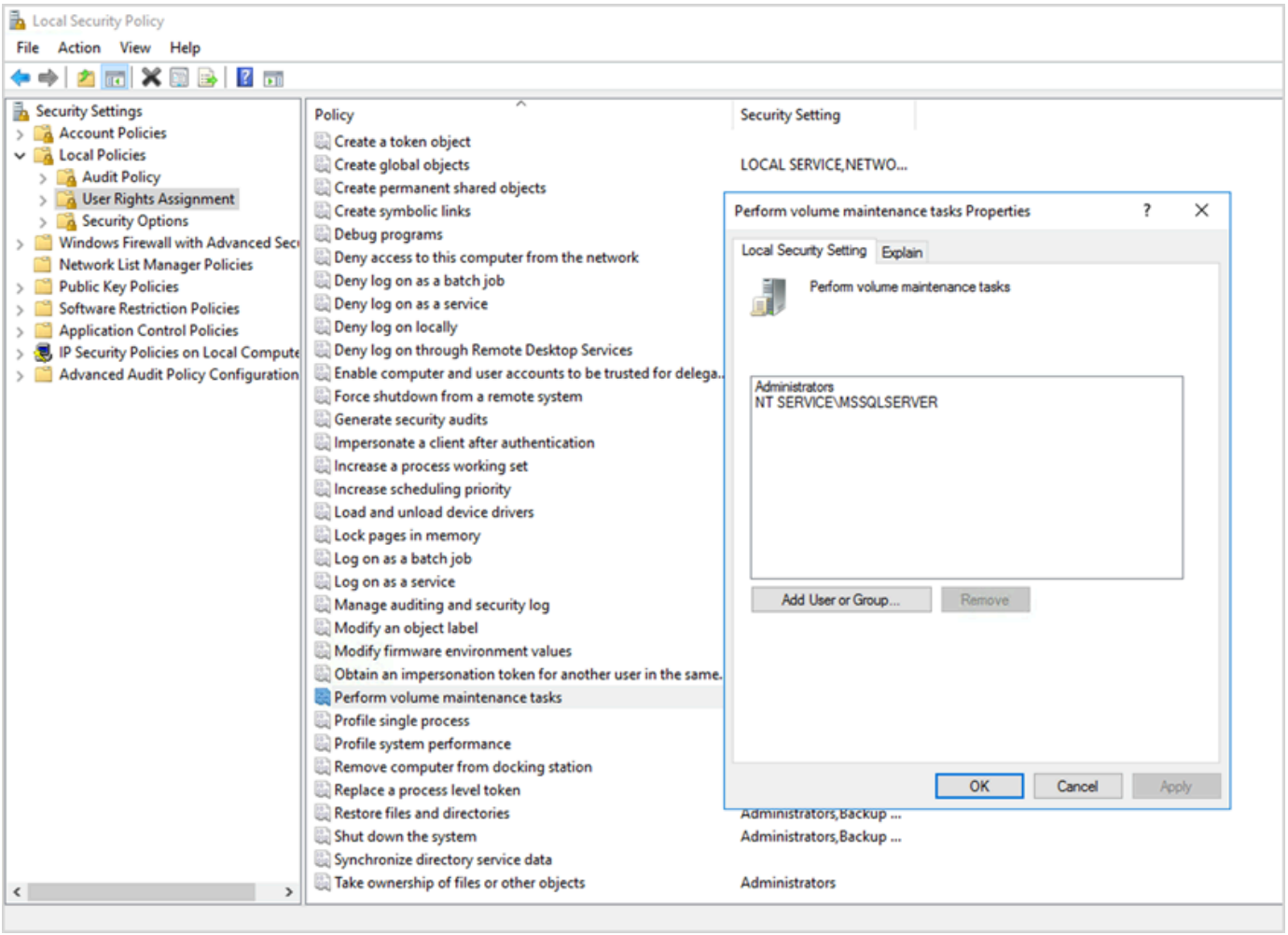
규제 제한을 따르지 않는 한 SQL Server에서는 볼륨 정비 작업 수행 설정을 사용하여 즉각적인 파일 초기화를 활성화하십시오. 이 옵션은 파일 자동 성장 성능을 크게 향상시킵니다.

이 설정은 데이터 파일의 제로화 작업을 건너뛰게 합니다. 즉, 데이터 파일은 초기화될 때 0 값(0x0)으로 채워지지 않으므로 시간이 오래 걸릴 수 있습니다. 디스크에 있는 현재 내용은 새 데이터가 디스크에 기록될 때만 덮어쓰여집니다.

> **Note**
>
> 로그 파일은 인스턴트 파일 초기화의 이점을 누릴 수 없습니다.

인스턴트 파일 초기화를 가능하게 하려면:

1. 시작 화면에서 `secpol.msc`를 실행하여 로컬 보안 정책 콘솔을 엽니다.
2. 다음 스크린샷과 같이 로컬 정책, 사용자 권한 할당, 볼륨 정비 작업 수행을 선택하고 SQL Server 서비스 계정을 추가합니다.



3. SQL Server 인스턴스를 다시 시작하면 변경 사항이 발효됩니다.

인스턴트 파일 초기화에 대한 자세한 설명은 Microsoft 웹사이트의 SQL Server 설명서를 참조하십시오.

> **보안 참고 사항**
>
> 인스턴트 파일 초기화를 사용하면 새 데이터가 파일에 기록될 때만 디스크를 덮어쓰므로 삭제된 콘텐츠를 읽을 수 있습니다.
> 드라이브가 인스턴스에 연결되어 있는 동안 파일에 있는 DACL(임의 액세스 통제 목록)은 SQL Server 서비스 계정 및 로컬 관리자에게만 액세스를 허용하므로 정보 공개 위험을 줄여줍니다. 하지만 파일을 분리하면 액세스할 수 있습니다. 삭제된 내용의 공개가 우려되는 경우, SQL Server 인스턴스의 인스턴트 파일 초기화를 비활성화해야 합니다.

메모리에서 페이지 잠그기

작동 시스템가 SQL Server 작업 세트를 트리밍하지 않도록 하려면 SQL Server 시작 계정의 메모리 내 페이지 잠금 옵션을 활성화하십시오.

이 옵션의 활성화 여부를 확인하려면 다음 SQL 쿼리를 사용하십시오:

```
SELECT sql_memory_model, sql_memory_model_desc
FROM sys.dm_os_sys_info;
```

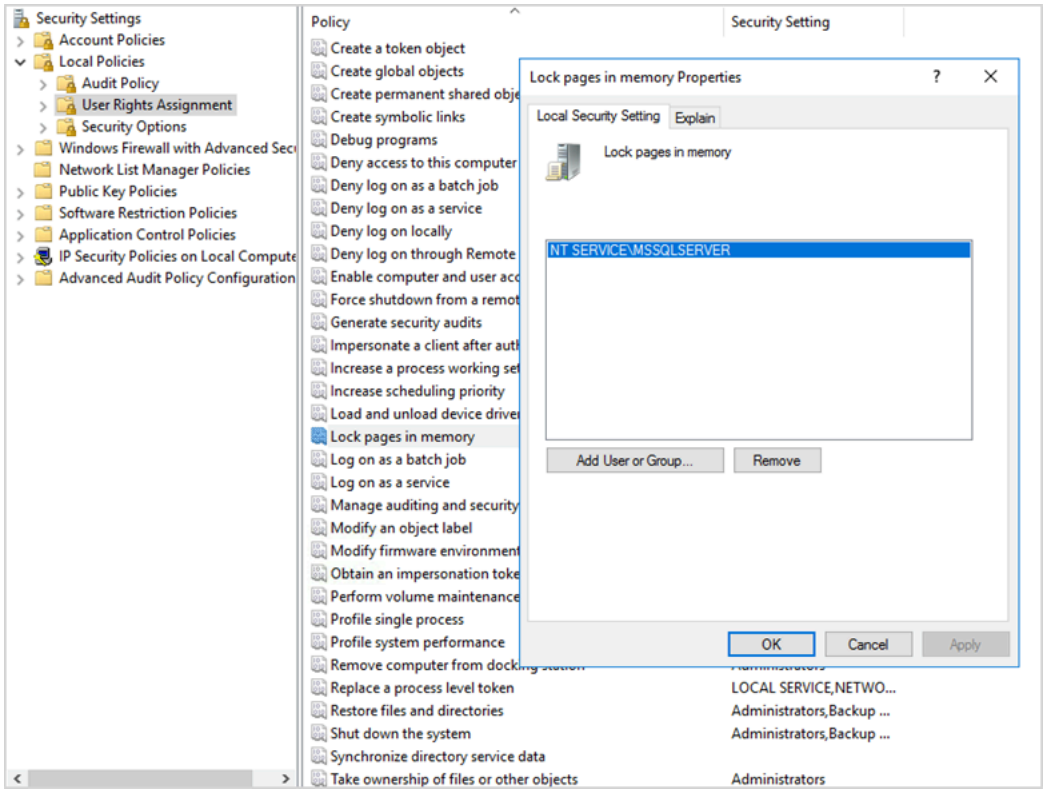
출력:

```
sql_memory_model     sql_memory_model_desc
1                   CONVENTIONAL

"CONVENTIONAL" means it’s not enabled.
```

메모리의 페이지 잠금 옵션을 활성화하려면:

1. 시작 화면에서 `secpol.msc`를 실행하여 로컬 보안 정책 콘솔을 엽니다.
2. 다음 스크린샷과 같이 로컬 정책, 사용자 권한 할당, 메모리의 페이지 잠금을 선택하고 SQL Server 서비스 계정을 추가합니다.



3. SQL Server 인스턴스를 다시 시작하면 변경 사항이 발효됩니다.
4. 다음 SQL 쿼리를 사용하여 메모리의 페이지 잠금 옵션이 활성화되었는지 확인합니다.

```
SELECT sql_memory_model, sql_memory_model_desc
FROM sys.dm_os_sys_info;
```

출력:

```
sql_memory_model     sql_memory_model_desc
2                    LOCK_PAGES

"LOCK_PAGES" means it's enabled.
```

SQL Server 메모리 모델에 대한 자세한 설명은 Microsoft 웹사이트의 sys.dm_os_sys_info 설명서에서 `sql_memory_model` 및 `sql_memory_model_desc`를 참조하십시오.

TCP 오프로드 및 RSS 설정 비활성

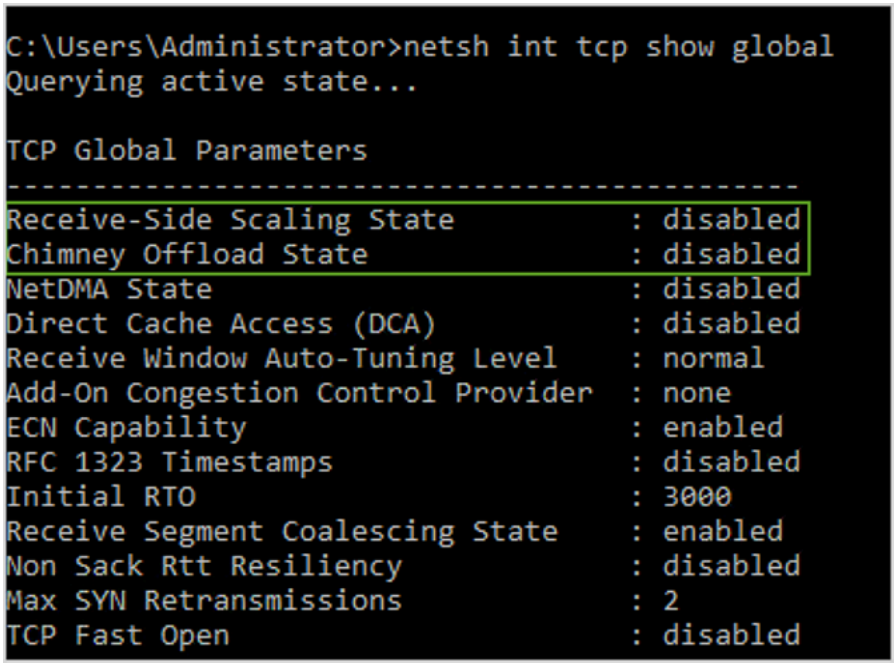
SQL 워크로드를 실행할 때 전송 수준 오류 또는 패킷 전송 오류와 같은 무작위 연결 문제가 발견되면 TCP 오프로드 및 RSS 설정을 비활성화하는 것이 좋습니다.

- TCP 오프로딩(TCP Chimney 오프로드 기능)은 프로세서에서 네트워크 어댑터로 TCP/IP 패킷 처리를 오프로드하여 CPU가 다른 작업에 사용할 수 있도록 합니다.
- 수신 측 스케일링(RSS)은 들어오는 네트워크 트래픽의 처리를 복수 프로세서 시스템에서 분산하는 데 도움이 됩니다. CPU 간에 네트워크 처리 부하를 효율적으로 분산합니다.

현재 설정을 확인하려면 명령 프롬프트에서 다음 netsh 명령을 실행하십시오:

```
$ netsh int tcp show global
```

아래는 이 명령에서 나온 출력의 예입니다. 이 예에서는 수신측 조정 상태와 Chimney 오프로드 상태가 모두 비활성화되어 있습니다.



특정 연결에 대한 작업 오프로드 정보를 가져오려면 명령 프롬프트에서 다음을 실행합니다:

```
netstat -t
```

그리고 오프로드 상태 열의 값을 확인합니다.

Windows Server 2008 및 2012에서 TCP 오프로딩 및 RSS를 사용하지 않도록 설정하려면 명령 프롬프트에서 다음 명령을 실행합니다:

```
netsh int ip set global taskoffload=disabled
netsh int tcp set global chimney=disabled
netsh int tcp set global rss=disabled
netsh int tcp set global netdma=disabled
```

이러한 설정에 대한 자세한 설명은 다음을 참조하세요:

- Microsoft 웹사이트의 TCP Chimney 오프로드, 수신측 스케일링, 네트워크 다이렉트 메모리 액세스 기능 및 수신측 조정 소개
- Amazon EC2 설명서의 TCP 오프로딩
- Amazon EC2 설명서의 PV 드라이버 문제 해결

> **Important**
>
> IPsec 태스크 오프로드 또는 TCP Chimney 오프로드를 사용하지 마십시오. Microsoft 설명서에 따르면 이러한 오프로드 기능은 Windows Server 2016에서 더 이상 사용되지 않으며 향후 버전에서는 지원되지 않을 수 있습니다. 이러한 기능을 사용하면 성능이 저하될 수 있습니다.

IOPS 및 처리량 요건 결정

Windows 성능 모니터를 사용하여 IOPS 및 처리량에 대한 정보를 얻을 수 있습니다.

Windows 성능 모니터를 열려면 명령 프롬프트에서 perfmon을 실행합니다. IOPS 및 처리량 데이터는 다음 성능 카운터에서 제공됩니다.

- 디스크 읽기/초 + 디스크 쓰기/초 = IOPS
- 디스크 읽기 바이트/초 + 디스크 쓰기 바이트/초 = 처리량

요건을 정확하게 예측하려면 피크 사용 시간과 일반적인 워크로드 주기 동안의 IOPS 및 처리량 데이터를 확보하는 것이 좋습니다. SQL Server용으로 선택한 인스턴스 타입이 이러한 I/O 요건을 지원하는지 확인하십시오.

이 추정치를 정확히 맞추는 것이 중요합니다. 그렇지 않으면 리소스를 과도하게 프로비저닝하여 리소스 활용도가 낮아지거나 리소스가 제대로 프로비저닝되지 않아 심각한 성능 문제가 발생할 수 있습니다.

스트라이핑을 사용하여 IOPS 및 처리량 제한을 우회할 수 있습니다.

SQL Server 애플리케이션에 EBS 볼륨에서 사용할 수 있는 최대 IOPS 및 처리량보다 많은 양이 필요한 경우, 이러한 제한을 극복하기 위해 EBS 볼륨을 스트라이핑하는 것을 고려해 보십시오.

스트라이핑 볼륨(RAID)은 IOPS 및 처리량 요건을 충족하는 데 도움이 되며 특정 인스턴스에서 지원하는 최대 IOPS 및 대역폭으로 제한됩니다. 스트라이핑 옵션에 대한 자세한 설명은 Amazon EC2 설명서의 RAID 구성을 참조하십시오. 독립형 서버에서도 스토리지 공간을 사용할 수 있습니다. 자세한 설명은 Microsoft 설명서를 참조하십시오.

바이러스 백신 소프트웨어에서 SQL Server 파일 제외

바이러스 백신 소프트웨어 설정을 구성할 때는 SQL Server 파일 및 디렉터리를 바이러스 검사에서 제외해야 합니다. 제외할 파일 및 디렉터리 목록에 대한 자세한 설명은 Microsoft 웹사이트에서 SQL Server를 실행하는 컴퓨터에서 실행할 바이러스 백신 소프트웨어를 선택하는 방법을 참조하십시오.

이러한 SQL Server 파일을 제외하지 않으면 SQL Server에서 해당 파일을 사용해야 할 때 바이러스 백신 소프트웨어에 의해 파일이 손상되거나 격리될 수 있습니다. 이러한 파일을 제외하지 않으면 성능 문제가 발생할 수도 있습니다.

SQL Server 구성

이 단원에서는 성능을 미세 조정하고, 일반적인 위험을 방지하고, 보안 및 가용성 요구 사항을 충족하도록 SQL Server 데이터베이스를 구성하는 모범 사례를 제공합니다. 데이터베이스를 Amazon EC2로 마이그레이션하기 전 또는 후에 이러한 변경 사항을 구현할 수 있습니다. 다음 단원에서는 구성 팁과 모범 사례를 제공합니다.

주제

경합을 줄이도록 tempdb 구성

크기가 같고 성장 계수가 동일한 여러 데이터 파일로 tempdb를 구성하는 것이 좋습니다.

tempdb를 많이 사용하는 사용량이 많은 데이터베이스 서버에서는 서버의 부하가 심하면 심각한 차단이 발생할 수 있습니다. 작업이 tempdb의 페이지를 가리키는 대기 리소스를 기다리는 것을 볼 수 있습니다. 이러한 페이지는 2:*x*:*x* 형식 (예: 2:1:1 또는 2:1:2)인 페이지 여유 공간 (PFS) 및 공유 글로벌 할당 맵(SGM) 페이지일 수 있습니다.

tempdb의 동시성을 높이려면 tempdb의 데이터 파일 수를 늘려 디스크 대역폭을 최대화하고 할당 구조에서의 경합을 줄일 수 있습니다. 여기에 몇 가지 지침이 있습니다.

- 논리적 프로세서 수가 8개 이하인 경우: 동일한 수의 데이터 파일 및 논리적 프로세서를 사용합니다.
- 논리 프로세서 수가 8개보다 많은 경우: 8개의 데이터 파일을 사용합니다.

경합이 계속되면 경합이 해결될 때까지 서버의 논리적 프로세서 수까지 데이터 파일 수를 4의 배수로 늘립니다. 이렇게 하면 tempdb에서 SGAM 경합을 방지하는 데 도움이 됩니다. SQL Server 2014 또는 이전 버전을 사용하는 경우 추적 플래그 1118도 활성화해야 합니다. 이 플래그는 혼합 익스텐트 대신 균일한 익스텐트로 페이지를 강제로 할당하므로 SGAM 페이지에서 스캔이 최소화되고 경합이 줄어듭니다.

SQL Server 2016(13.x)부터는 이 동작이 `ALTER DATABASE`의 `AUTOGROW_SINGLE_FILE` 및 `AUTOGROW_ALL_FILES` 옵션을 통해 제어됩니다. 예:

```
alter database <database name> MODIFY FILEGROUP [PRIMARY] AUTOGROW_ALL_FILES
```

이러한 옵션 설정에 대한 자세한 내용은 Microsoft SQL Server 설명서를 참조하세요.

최상의 성능을 위한 MAXDOP 설정

최대 병렬 처리(MAXDOP)는 여러 CPU에서 SQL Server를 실행하기 위한 서버 구성 옵션입니다. 이는 병렬 계획 실행에서 단일 명령문을 실행하는 데 사용되는 프로세서 수를 제어합니다. 기본값은 0이며, 이를 통해 SQL Server는 사용 가능한 모든 프로세서를 사용할 수 있습니다. 이는 성능에 영향을 줄 수 있으며 대부분의 사용 사례에는 적합하지 않습니다.

SQL Server의 MAXDOP 값을 구성할 때는 다음 지침을 따르세요.

NUMA 노드	논리적 프로세서	MAXDOP 값
단일	≤ 8	4개, 2개 또는 코어 수(1개 또는 2개 코어용)
단일	> 8	8, 4 또는 2
다중	≤ 16	8, 4 또는 2
다중	> 16	16, 8, 4 또는 2

> **Note**
>
> 일반적으로 MAXDOP를 2, 4 또는 8로 설정하면 대부분의 사용 사례에서 최상의 결과를 얻을 수 있습니다. 워크로드를 테스트하고 `CXPACKET` 같은 병렬 처리 관련 대기 유형이 있는지 모니터링하는 것이 좋습니다.

다음 쿼리를 사용하여 SQL Server 2016 및 이후 버전의 현재 NUMA 구성을 수집할 수 있습니다.

```
select @@SERVERNAME,
SERVERPROPERTY('ComputerNamePhysicalNetBIOS'),
cpu_count,
hyperthread_ratio,
softnuma_configuration,
softnuma_configuration_desc,
socket_count,
numa_node_count
from
sys.dm_os_sys_info
```

여기서 각 항목은 다음과 같습니다.

- `cpu_count`는 시스템에 있는 논리적 CPU의 수를 나타냅니다.
- `hyperthread_ratio`는 물리적 프로세서 1개가 노출하는 코어 수의 비율입니다.
- `softnuma_configuration`이 `0`, 1 또는 2:
  - `0 (OFF)`: default
  - `1 (automated)`: 소프트 NUMA
  - `2 (manual)`: 소프트 NUMA
- `softnuma_configuration_desc`가 `OFF`, `ON` 또는 `MANUAL`:
  - `OFF`는 소프트 NUMA 기능이 꺼져 있음을 나타냅니다.
  - `ON`은 SQL Server에서 NUMA 노드 크기를 자동으로 결정함을 나타냅니다.
  - `MANUAL`은 소프트 NUMA가 수동으로 구성되었음을 나타냅니다.
- `socket_count`는 프로세서 소켓 수입니다.
- `numa_node_count`는 시스템에서 사용 가능한 NUMA 노드 수입니다.

현재 MAXDOP 값을 확인하려면 다음을 사용하세요.

```
$ sp_configure 'max_degree_of_parallelism'
```

사용자 이름에 대한 자세한 정보는 Microsoft SQL 설명서를 참조하세요.

병렬 처리에 대한 비용 임계값 변경

병렬 처리의 비용 임계값에 따라 병렬 실행에 적합한 쿼리가 결정됩니다. 이 속성의 기본값은 5입니다. 즉, 직렬 계획의 비용이 5 (예상 시간이 아닌 추상화된 비용 단위를 나타냄) 를 초과하는 경우 최적화 프로그램이 병렬 계획으로 전환합니다. 이 속성을 더 높은 수로 설정하는 것이 좋습니다.

프로세서의 가격이 비싸고 처리 능력이 낮으며 쿼리 처리 속도가 지금보다 느렸을 때는 기본값이 적절했습니다. 오늘날의 프로세서는 훨씬 더 빠릅니다. 따라서 비교적 작은 쿼리(예: 비용 임계값이 32인 경우)는 병렬 실행으로 큰 이점을 얻지 못합니다. 특히 병렬 실행 조정과 관련된 오버헤드를 고려할 때 그렇습니다.

대부분의 경우 병렬 처리의 비용 임계값을 50으로 설정하는 것이 좋습니다. 다음은 병렬 처리의 비용 임계값을 구성하는 방법을 보여주는 예입니다.

```
USE sampledb;
GO
EXEC sp_configure 'show advanced options', 1 ;
GO
RECONFIGURE
GO
EXEC sp_configure 'cost threshold for parallelism', 50 ;
GO
RECONFIGURE
GO
```

임시 워크로드에 대한 최적화

임시 워크로드에 최적화 옵션을 활성화하면 일회용 임시 배치가 많이 포함된 워크로드에 대한 계획 캐시의 효율성을 개선할 수 있습니다. 임시 쿼리를 처음 실행하면 데이터베이스 엔진이 전체 실행 계획 대신 컴파일된 계획 스텁을 캐시하므로 계획 캐시의 공간이 절약됩니다. 임시 배치를 다시 실행하면 데이터베이스 엔진이 해당 배치가 이전에 실행된 것으로 인식하고 컴파일된 계획 스텁을 계획 캐시에 있는 컴파일된 전체 계획으로 바꿉니다.

이 옵션의 활성화 여부를 확인하려면 다음 쿼리를 사용하세요.

```
$ sp_configure 'optimize for ad hoc workloads'
```

임시 워크로드를 최적화하는 방법에 대한 자세한 내용은 Microsoft SQL Server 설명서를 참조하세요.

추적 플래그를 사용하여 성능 향상

성능을 향상시키기 위해 환경에 적용할 수 있는 SQL Server 추적 플래그를 사용하는 것을 고려해 보세요. 예:

- 4199: SQL Server 누적 업데이트(CU) 및 서비스 팩(SP)에서 릴리스된 쿼리 옵티마이저(QO) 변경을 활성화합니다.
- 8048: NUMA로 분할된 메모리 객체를 CPU로 분할된 메모리 객체로 변환합니다.
- 9024: 글로벌 로그 풀 메모리 객체를 NUMA로 분할된 메모리 객체로 변환합니다.

다음 예는 Amazon EC2의 SQL Server에 대한 추적 플래그를 켜고 끄는 방법을 보여줍니다. 추적을 활성화할 때 문제가 발생하는 경우 계정에 대한 적절한 권한이 있는지 확인하세요.

추적 플래그 4199를 켜려면 다음을 실행합니다.

```
dbcc traceon (4199, -1);
```

추적 플래그의 상태를 확인하려면 다음을 실행합니다.

```
dbcc tracestatus (4199);
```

추적 플래그 4199를 끄려면 다음을 실행합니다.

```
dbcc traceoff (4199, -1);
dbcc tracestatus (4199);
```

추적 플래그의 전체 목록은 Microsoft SQL Server 설명서를 참조하세요.

최신 패치를 설치하세요.

Microsoft는 SQL Server 2017부터 서비스 팩(SP)의 출시를 중단했습니다. 누적 업데이트(CU) 및 중요 업데이트(GDR)만 릴리스됩니다.

SP에는 SQL Server의 중요한 수정 사항이 포함되어 있으므로 최신 SP가 설치되어 있는지 확인하세요. 또한 가능하면 최신 CU 패키지를 설치하세요.

최신 SQL Server 업데이트에 대한 자세한 내용은 Microsoft 웹 사이트에서 Microsoft SQL Server의 최신 업데이트를 참조하세요.

메모리 부족을 피하기 위한 최대 서버 메모리 제한

성능상의 이유로 SQL Server는 이미 할당한 메모리를 릴리스하지 않습니다. SQL Server가 시작되면 min_server_memory 옵션에 지정된 메모리를 천천히 사용하다가 max_server_memory 옵션에 지정된 값에 도달할 때까지 계속 늘립니다. (이러한 설정에 대한 자세한 내용은 SQL Server 설명서의 Server 메모리 구성 옵션을 참조하세요.)

SQL Server 메모리에는 버퍼 풀과 비버퍼 풀(memory to leave 또는 MTL이라고도 함)이라는 두 가지 구성 요소가 있습니다. max_server_memory 옵션의 값은 버퍼 캐시, 프로시저 캐시, 계획 캐시, 버프 구조 및 기타 캐시로 구성된 SQL Server 버퍼 풀의 크기를 결정합니다.

SQL Server 2012부터 min_server_memory 및 max_server_memory는 `SQLGENERAL`, `SQLBUFFERPOOL`, `SQLQUERYCOMPILE`, `SQLQUERYPLAN`, `SQLQUERYEXEC`, `SQLOPTIMIZER`, `SQLCLR`을 포함하여 모든 캐시에 대한 모든 메모리 할당을 고려합니다. max_server_memory의 전체 메모리 클럭 목록은 Microsoft SQL Server 설명서의 sys.dm_os_memory_clerks를 참조하세요.

현재 max_server_memory 값을 확인하려면 다음 명령을 사용하세요.

```
$ sp_configure 'max_server_memory'
```

max_server_memory는 시스템 전체의 메모리 압력을 유발하지 않는 값으로 제한하는 것이 좋습니다. 모든 환경에 적용되는 보편적인 공식은 없지만 이 단원에서는 몇 가지 지침을 제공했습니다. max_server_memory는 동적 옵션이므로 런타임에 변경할 수 있습니다.

시작점으로 max_server_memory를 다음과 같이 결정할 수 있습니다.

```
max_server_memory = total_RAM - (memory_for_the_OS + MTL)
```

여기서 각 항목은 다음과 같습니다.

- 운영 체제의 메모리는 1~4GB입니다.
- MTL(memory to leave)에는 64비트 시스템에서 워커 스레드당 2MB인 스택 크기가 포함되며, 이 크기는 다음과 같이 계산할 수 있습니다. `MTL = stack_size * max_worker_threads`

또는 다음을 사용할 수 있습니다.

```
max_server_memory = total_RAM - (1 GB for the OS
 + memory_basis_amount_of_RAM_on_the_server)
```

여기서 RAM의 메모리 기본 용량은 다음과 같이 결정됩니다.

- 서버의 RAM이 4GB에서 16GB 사이인 경우 RAM 4GB당 1GB를 남겨 두세요. 예를 들어, 용량이 16GB인 서버의 경우 4GB를 남겨 두세요.
- 서버의 RAM이 16GB를 초과하는 경우, 16GB까지는 RAM 4GB당 1GB를, 16GB 초과의 RAM에는 8GB당 1GB를 남겨 두세요.

예를 들어, 서버의 RAM이 256GB인 경우 계산은 다음과 같습니다.

- 운영 체제의 경우 1GB
- 최대 16GB RAM: 16/4 = 4GB
- 16GB를 초과하는 남은 RAM: (256-16)/8 = 30
- 남은 총 RAM: 1 + 4 + 30 = 35GB
- max_server_memory: 256 - 35 = 221GB

초기 구성 후에는 일반적인 워크로드 기간 동안 확보할 수 있는 메모리를 모니터링하여 SQL Server에 할당된 메모리를 늘리거나 줄여야 하는지 판단하세요.

> **Note**
>
> Windows는 96MB에서 메모리 리소스 부족 알림을 보내므로 버퍼가 필요하지만 RAM이 256GB 이상인 대규모 서버의 경우 사용 가능한 MB를 1GB 초과로 설정할 수 있습니다.

자세한 내용은 Microsoft SQL Server 설명서의 메모리 관리 아키텍처 설명서를 참조하세요.

가장 높은 데이터베이스 호환성 수준 사용

SQL Server의 최신 개선 사항을 활용하려면 현재 데이터베이스 호환성 수준을 사용하고 있는지 확인하세요. 데이터베이스를 하위 버전에서 상위 버전으로 복원할 때 하위 버전의 호환성 수준이 유지되므로 이 점을 확인하는 것이 중요합니다. 최신 데이터베이스 개선 사항 중 일부는 데이터베이스 호환성을 설치한 엔진 버전에서 사용할 수 있는 최신 수준으로 설정한 경우에만 유효합니다.

현재 데이터베이스 호환성을 확인하려면 다음을 사용하세요.

```
$ select name, compatibility_level from sys.databases
```

데이터베이스 호환성 수준에 대한 자세한 내용은 Microsoft SQL Server 설명서를 참조하세요.

VLF 수 제어

최대 데이터 크기 및 로그 파일을 미리 할당하세요. 성능을 높이려면 공간을 미리 할당하고 로그 파일의 자동 증가(자동 증가) 설정을 수정하여 가상 로그 파일(VLF)의 수를 제어하세요.

일반적으로 8GB의 자동 증가 계수는 대부분의 프로덕션 환경에서 잘 작동합니다. 트랜잭션 로그 파일을 8GB 단위로 늘리는 것을 고려해 보세요. VLF 수가 많으면 데이터베이스의 백업 및 복구 시간이 늘어나고 로그 파일을 거쳐야 하는 모든 작업(예: 복제)에서 성능 문제가 발생할 수 있습니다.

VLF 생성 및 증가 알고리즘에 대한 자세한 내용은 SQLskills 블로그를 참조하세요.

데이터베이스 자동 증가 설정 확인

데이터 또는 로그 파일을 늘려야 하는 모든 트랜잭션에는 파일 증가 작업에 소요되는 시간이 포함됩니다. 파일은 FILEGROWTH 옵션에 정의된 증분 크기만큼 늘어납니다. SQL Server 프로파일러 추적에서 파일 증가 이벤트를 찾을 수 있습니다. 파일 증가에 시간이 오래 걸리는 경우 데이터 처리 속도가 매우 느릴 때 발생하는 ASYNC_IO_COMPLETION 같은 대기 유형이 나타날 수 있습니다. 이러한 대기 유형은 성능에 영향을 줄 뿐만 아니라 트랜잭션 시간 초과를 초래할 수도 있습니다. 해당 트랜잭션이 다른 트랜잭션에서 찾는 리소스에 대한 잠금을 유지하는 경우 시간 초과로 인해 심각한 서버 차단 문제가 발생할 수 있습니다.

따라서 자동 증가 설정을 매우 신중하게 구성하는 것이 좋습니다. 다음과 같은 사항에 유의하세요.

- 파일 증가는 SQL Server에서 가장 비용이 많이 드는 작업 중 하나입니다.
- 작은 청크가 자주 자동 증가하면 디스크 조각화가 발생할 수 있습니다.
- 이전 단원에서 설명한 것처럼 로그 파일이 자주 자동 증가하면 가상 로그 파일(VLF)의 수가 많아지고 성능에 영향을 미칩니다.

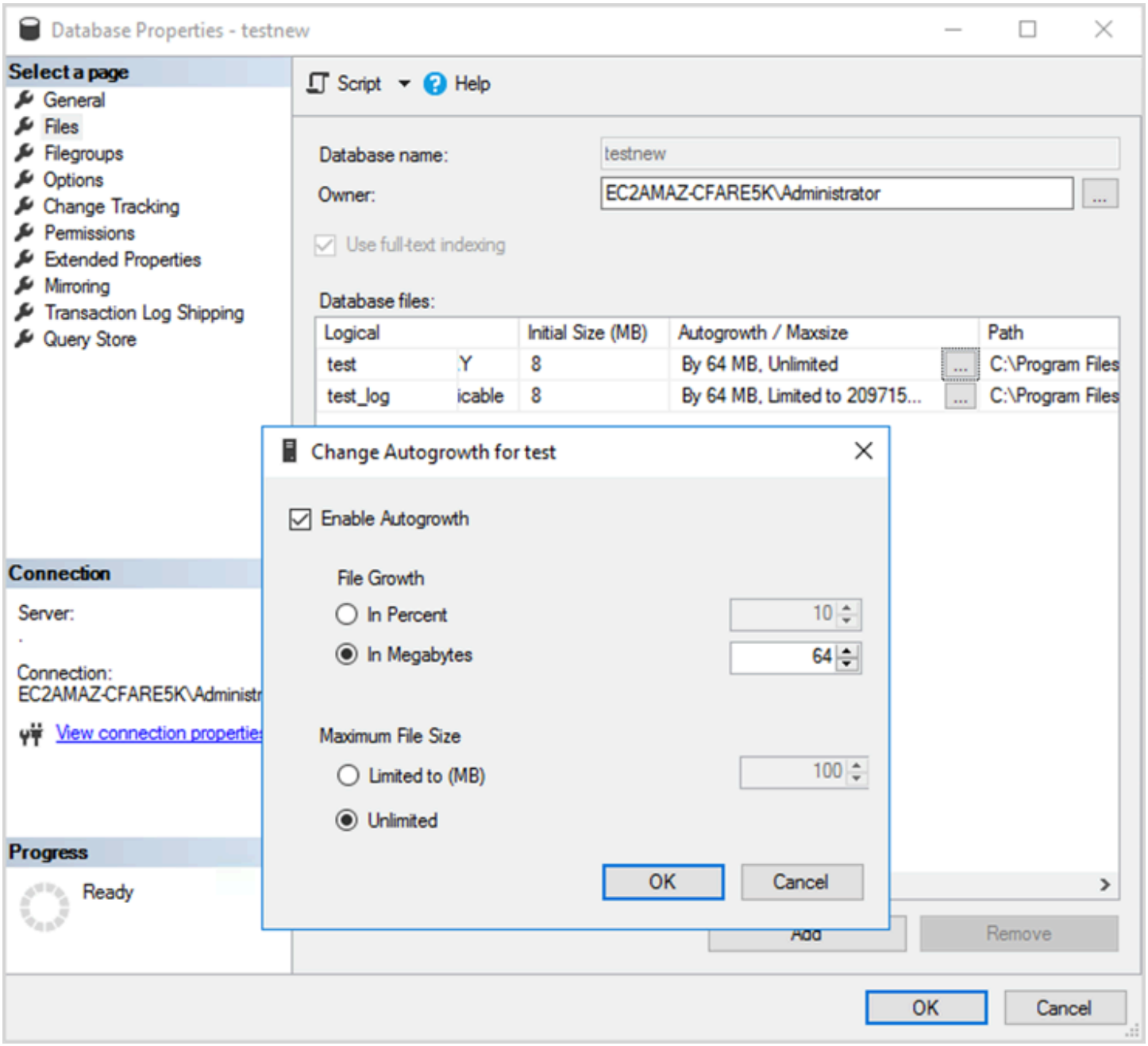
이러한 모든 이유로 인해 데이터베이스 시작 속도가 느려지고 백업 및 복구 시간이 늘어날 수 있습니다.

이상적으로는 정기적인 모니터링을 기반으로 사전에 파일을 미리 확장하는 것이 좋습니다. 자동 증가를 백분율로 설정하거나 정적 값(MB)으로 설정하는 것 중에서 신중하게 선택하세요. 일반적으로 자동 증가를 파일 크기의 8분의 1로 설정하는 것이 좋지만 이 방법은 올바른 선택이 아닐 수 있습니다. (예를 들어, 데이터 파일 크기가 몇 TB인 경우 이 비율은 너무 높을 수 있습니다.)

대부분의 경우, 대부분의 대형 데이터베이스의 데이터 파일에는 1024MB의 자동 증가 값이 적합합니다. 로그 파일의 경우 512MB로 시작하는 것이 좋습니다. 비상 조치를 취하려면 자동 증가 값을 설정하는 것이 좋지만 과거 추세를 기반으로 파일을 몇 개월 동안 수동으로 늘리는 것이 좋습니다.

> **Note**
>
> 자동 증가 설정은 비상 조치여야 하므로 파일에 스토리지를 미리 할당한 후에 설정해야 합니다.

SQL Server Management Studio(SSMS) 또는 Transact-SQL을 사용하여 자동 증가 설정을 변경할 수 있습니다. 다음 화면 그림은 SSMS의 자동 증가 설정을 보여줍니다.



데이터 및 로그 파일에 FILEGROWTH 옵션을 사용하는 경우 백분율로 설정하거나 정적 값(MB)으로 설정하는 것 중에서 신중하게 선택하세요. 백분율을 설정하면 파일 크기가 계속 증가하므로 증가율을 더 잘 제어하려면 정적 크기를 사용하는 것이 좋습니다.

- SQL Server 2022(16.x) 이전 버전에서는 트랜잭션 로그에서 즉각적인 파일 초기화를 사용할 수 없으므로 로그 증가 시간 연장이 특히 중요합니다.
- SQL Server 2022(16.x, 모든 에디션)부터는 즉각적인 파일 초기화를 통해 최대 64MB의 트랜잭션 로그 증가 이벤트에 도움이 될 수 있습니다. 새 데이터베이스의 기본 자동 증가 크기 증가는 64MB입니

다. 64MB보다 큰 트랜잭션 로그 파일 자동 증가 이벤트는 즉각적인 파일 초기화의 이점을 누릴 수 없습니다.

Always On 가용 그룹 구성

SQL Server 버전 2012 이상용 네이티브 클라이언트 라이브러리와 .NET Framework 4.5 라이브러리를 사용하는 경우 MultiSubnetFailover 매개변수를 사용하여 연결 동작을 변경할 수 있습니다. 이 파라미터를 TRUE으로 설정하길 권장합니다. 이렇게 하면 Always On 가용성 그룹을 사용하여 장애 조치 속도를 높일 수 있습니다.

> **Note**
>
> MultiSubnetFailover 매개변수를 사용할 수 없는 레거시 애플리케이션이 있는 경우 SQL Server 인스턴스 앞에 Network Load Balancer를 배치할 수 있습니다. 밸런서는 상태 점검을 사용하여 어떤 SQL Server 데이터베이스가 활성 상태인지 확인하고 현재 해당 데이터베이스를 호스팅하는 인스턴스로 트래픽을 전송합니다. 로드 밸런서는 하나 이상의 가용 영역에 걸쳐 있습니다. 상태 점검에 59999와 같은 전용 포트를 사용한 다음 해당 포트에 응답하도록 클러스터 그룹 매개변수를 수정할 수 있습니다. 이렇게 하면 MultiSubnetFailover 매개변수를 사용하지 않고도 SQL Server 장애 조치 시간을 약 1분으로 줄일 수 있습니다. 자세한 지침은 블로그 게시물 Network Load Balancer를 사용하여 Amazon EC2 인스턴스에서 SQL Server의 장애 조치 시간 줄이기를 참고하십시오.

가용 그룹 리스너가 DNS에 등록되는 방식에 영향을 미치는 두 가지 설정: RegisterAllProvidersIP 및 HostRecordTTL.

Always On 가용 그룹을 사용할 때는 RegisterAllProvidersIP를 true로 설정합니다.

RegisterAllProvidersIP는 1(true)로 설정하는 것을 권장합니다. RegisterAllProvidersIP를 1로 설정하여 가용 그룹 리스너를 만들면 해당 리스너의 모든 IP 주소가 DNS에 등록됩니다. RegisterAllProvidersIP를 0(false)으로 설정하면 활성 IP가 하나만 등록됩니다.

장애 조치의 경우 기본 복제본이 한 서브넷에서 다른 서브넷으로 이동할 때 이전 IP 주소가 등록 취소되고 새 IP 주소가 등록됩니다. 가용 그룹 리스너가 온라인 상태가 되면 DNS가 새 IP로 업데이트됩니다. 하지만, 클라이언트 시스템은 현재 캐시된 항목이 만료될 때까지 리스너 이름을 새 IP 주소로 귀결하지 않습니다.

Always On 가용 그룹을 사용할 때는 HostRecordTTL을 60% 이하로 설정하십시오.

HostRecordTTL 설정은 캐시된 DNS 항목에 대한 Time to Live(TTL)를 제어합니다. 기본 값은 1200초입니다. HostRecordTTL을 훨씬 더 낮은 설정(60초 이하)으로 변경하는 것을 권장합니다. 이렇게 하면 캐시된 값이 더 빨리 만료되므로 장애 조치 시 클라이언트 시스템이 새 IP를 더 빨리 귀결할 수 있습니다.

Always On 클러스터 그룹의 자동 장애 복구를 비활성화합니다.

Windows Cluster Manager의 Always On 가용 그룹에 대해 자동 장애 복구를 사용하지 않도록 설정했는지 확인하십시오.

백업 구성

디스크 레이아웃 또는 파일 배포 최적화 섹션에서 설명한 대로 기본 SQL Server 백업을 별도의 드라이브로 보내는 것이 좋습니다. 또한 백업 파일이 있는 EBS 볼륨의 예약된 스냅샷을 만드는 것도 고려해 보세요.

데이터베이스 최적화 개선하기

이 단원에서는 SQL Server 쿼리 옵티마이저를 사용할 때 성능을 개선하기 위한 모범 사례를 제공합니다. 인덱스를 다시 구축하고 테이블 통계를 정기적으로 업데이트하면 실행 계획을 최적화하는 데 어떻게 도움이 되는지 설명합니다. 다음 단원에서는 구성 팁과 모범 사례를 제공합니다.

주제

인덱스 재구축

쿼리 옵티마이저가 최상의 쿼리 계획을 생성하고 올바른 인덱스를 사용할 수 있도록 하려면 인덱스가 조각화되어서는 안 됩니다. 인덱스는 업데이트, 삽입 또는 삭제율에 기반하여 시간이 지남에 따라 조각화됩니다. 정기적으로 테이블을 리인덱싱해야 합니다. 재구축 빈도는 데이터베이스가 DML(데이터 조작 언어) 작업을 처리하는 속도에 따라 달라집니다.

30% 이상 조각화된 인덱스는 다시 구축하고 30% 미만으로 조각화된 인덱스는 다시 구성하는 것이 좋은 시작점이 될 것입니다. 30%라는 값은 대부분의 사용 사례에서 효과가 있지만, 사용하지 않은 인덱스로 인해 쿼리 계획이 여전히 좋지 않은 것이 확인된다면 이 비율을 다시 살펴봐야 할 수도 있습니다.

다음과 같은 쿼리를 사용하여 조각화를 확인하십시오.

```
SELECT OBJECT_NAME(OBJECT_ID), index_id,index_type_desc,index_level,
avg_fragmentation_in_percent,avg_page_space_used_in_percent,page_count
FROM sys.dm_db_index_physical_stats
(DB_ID(N'<your_database>'), NULL, NULL, NULL , 'SAMPLED')
ORDER BY avg_fragmentation_in_percent DESC
```

정기적으로 인덱스를 재구축하기 위한 유지 관리 작업을 생성하는 것을 권장합니다.

UPDATE STATISTICS

조각화된 인덱스와 마찬가지로 옵티마이저에 테이블 열의 키 값(통계) 분포에 대한 최신 정보가 없으면 최적의 실행 계획을 생성할 수 없습니다. 정기적으로 모든 테이블의 통계를 업데이트하는 것을 권장합니다. 업데이트 빈도는 데이터베이스가 DML 작업을 처리하는 속도에 따라 다르지만 일반적으로 사용량이 적은 시간에는 일주일에 두 번 실행됩니다. 하지만, 인덱스를 다시 구축하는 날에는 통계를 업데

이트하지 마십시오. 자세한 내용은 Microsoft 설명서에서 Microsoft SQL Server 에이전트를 참조하세요.

데이터베이스 최적화를 위해 인덱스 및 통계 유지 관리 스크립트를 사용하는 것을 권장합니다. 예제는 SQL Server 유지 관리 솔루션 웹사이트에 제공된 SQL Server 인덱스 및 통계 유지 관리 스크립트를 참고하십시오.

AWS Launch Wizard를 사용하여 Amazon EC2에서의 SQL Server 배포 최적화

AWS Launch Wizard는 Amazon EC2에서 SQL Server 단일 인스턴스 및 고가용성(HA) 배포를 위한 기본 방법입니다. Launch Wizard 배포는 AWS Well-Architected 프레임워크를 기반으로 하며 보안, 신뢰성, 성능 효율성 및 비용 절감을 위해 최적화되어 있습니다.

Launch Wizard를 사용하면 SQL Server 배포를 간소화하고 SQL Server를 더 쉽게 구성할 수 있습니다. 이 기능은 다음과 같습니다.

- 자동 AWS 리소스 선택 — Launch Wizard는 가상 CPU(vCPU), 메모리 및 네트워킹 요구 사항을 기반으로 최적의 인스턴스 유형을 추천할 수 있습니다. 또한 스토리지 드라이브 및 처리량을 기반으로 볼륨 유형을 추천할 수 있습니다.
- 원클릭 모니터링 — Launch Wizard는 Amazon CloudWatch Application Insights와 통합되어 AWS에서 SQL Server HA 배포에 대한 모니터링을 설정합니다. 이 옵션을 선택하면 Application Insights는 CloudWatch에서 관련 지표, 로그, 경보를 자동으로 설정하고 새로 배포된 워크로드의 모니터링을 시작합니다.
- 쉽게 검색할 수 있는 애플리케이션 리소스 그룹 - Launch Wizard는 SQL Server 애플리케이션용으로 만든 모든 AWS 리소스에 대한 리소스 그룹을 만듭니다. AWS Systems Manager 콘솔에서 SQL Server 애플리케이션을 관리, 패치 및 유지 관리할 수 있습니다.

Launch Wizard는 재사용 가능한 AWS CloudFormation 코드 템플릿을 제공합니다. 이러한 템플릿은 후속 애플리케이션 배포의 기준으로 사용할 수 있습니다. 자세한 내용은 AWSLaunch Wizard 개요 및 사용 설명서를 참조하세요.

다음 단계

이 설명서에서는 Amazon EC2에서 Microsoft SQL Server 워크로드를 구성하고 실행하기 위한 몇 가지 모범 사례를 다루었습니다. 마이그레이션 프로세스의 계획 및 구현 단계에서 이러한 지침을 따르면 프로덕션 환경에서 안정적인 서버를 구축하는 데 도움이 됩니다.

이러한 구성 작업에 대한 자세한 내용은 각 단원에 제공된 링크를 참조하고 추가 리소스 단원에 나열된 웹 페이지를 방문하십시오.

추가 리소스

관련 전략, 가이드 및 패턴

- Migration strategy for relational databases
- SQL Server 패턴:
  - 모든 패턴
  - 리호스팅 패턴(SQL 서버를 Amazon EC2로 마이그레이션)
  - 리플랫폼 패턴(SQL Server를 Amazon RDS for SQL Server로 마이그레이션)
  - 리아키텍트 패턴(SQL Server를 오픈 소스 및 AWS 클라우드 네이티브 데이터베이스로 마이그레이션)
- AWS 권장 가이드 웹사이트

AWS 리소스

- AWS 설명서
- AWS 일반 참조
- AWS 용어집

AWS 서비스

- Amazon EBS
- Amazon EC2

기타 리소스

- EBS 볼륨이 Windows Server 2016 이상 AMI에서 초기화를 수행하지 않음
- Microsoft SQL Server tempdb를 Amazon EC2의 인스턴스/임시 디스크로 이동하는 방법
- 시작 시 Windows 인스턴스에서 명령 실행
- 시작 시 Windows 임시 디스크 스트라이프
- HammerDB를 사용한 SQL Server 벤치마킹
- 내 SQL Server에 실제로 필요한 메모리 용량은?
- SQL Server 대기 통계 (또는 어디에 문제가 있는지 알려주십시오...)

- 다중 서브넷 가용성 그룹의 연결 시간 초과
- 캐시를 계획하고 임시 워크로드에 맞게 최적화
- Windows에서 RAM, 가상 메모리, 페이지파일 및 메모리 관리
- 64비트 버전의 Windows에 적합한 페이지 파일 크기를 결정하는 방법

문서 기록

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

변경 사항	설명	날짜
수정된 정보	병렬 처리 속성의 비용 임계값에 대한 정보를 수정했습니다. 값은 시간이 아닌 비용 단위로 측정됩니다.	2023년 12월 4일
업데이트된 지침	NTFS 할당 단위 크기 설정, 메모리의 페이지 잠금, 작업 오프로드 기능 사용, 스트라이핑 사용, 병렬 처리 비용 임계값 변경, 추적 플래그 사용, 데이터베이스 자동 확장 설정 사용에 대한 섹션이 업데이트되었습니다.	2023년 8월 8일
추가된 지침	파라미터를 사용할 수 없는 레거시 애플리케이션에 Network Load Balancer를 사용하는 방법에 대한 정보가 추가되었습니다. MultiSubnetFailover	2022년 11월 11일
코드가 수정되었습니다.	인스턴스 스토어 초기화 섹션의 PowerShell 코드를 수정했습니다.	2022년 6월 27일
새 섹션 추가됨	SQL Server용 AWS Launch Wizard에 대한 정보가 추가되었습니다.	2021년 8월 18일
최초 게시	—	2020년 7월 21일

AWS 권장 가이드 용어집

다음은 AWS 권장 가이드에서 제공하는 전략, 가이드, 패턴 등에서 일반적으로 사용되는 용어입니다. 용어집 항목을 제안하려면 용어집 끝에 있는 피드백 제공 링크를 사용하세요.

숫자

7가지 전략

애플리케이션을 클라우드로 이전하기 위한 7가지 일반적인 마이그레이션 전략 이러한 전략은 Gartner가 2011년에 파악한 5가지 전략을 기반으로 하며 다음으로 구성됩니다.

- 리팩터링/리아키텍트 - 클라우드 네이티브 기능을 최대한 활용하여 애플리케이션을 이동하고 해당 아키텍처를 수정함으로써 민첩성, 성능 및 확장성을 개선합니다. 여기에는 일반적으로 운영 체제와 데이터베이스 이식이 포함됩니다. 예를 들어, 온프레미스 Oracle 데이터베이스를 Amazon Aurora PostgreSQL 호환 버전으로 마이그레이션합니다.
- 리플랫포밍(리프트 앤드 리셰이프) - 애플리케이션을 클라우드로 이동하고 일정 수준의 최적화를 도입하여 클라우드 기능을 활용합니다. 예를 들어, 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 Amazon RDS for Oracle로 마이그레이션합니다.
- 재구매(드롭 앤드 숍) - 일반적으로 기존 라이선스에서 SaaS 모델로 전환하여 다른 제품으로 전환합니다. 예를 들어, 고객 관계 관리(CRM) 시스템을 Salesforce.com으로 마이그레이션합니다.
- 리호스팅(리프트 앤드 시프트) - 애플리케이션을 변경하지 않고 클라우드로 이동하여 클라우드 기능을 활용합니다. 예를 들어, 온프레미스 Oracle 데이터베이스를 AWS 클라우드의 EC2 인스턴스에 있는 Oracle로 마이그레이션합니다.
- 재배치(하이퍼바이저 수준의 리프트 앤 시프트) - 새 하드웨어를 구매하거나, 애플리케이션을 다시 작성하거나, 기존 운영을 수정하지 않고도 인프라를 클라우드로 이동합니다. 이 마이그레이션 시나리오는 온프레미스 환경과 AWS 간의 가상 머신 호환성 및 워크로드 이동성을 지원하는 AWS의 VMware Cloud에 한정됩니다. 인프라를 AWS의 VMware Cloud로 마이그레이션할 때 온프레미스 데이터 센터에서 VMware Cloud Foundation 기술을 사용할 수 있습니다. 예를 들어, Oracle 데이터베이스를 호스팅하는 하이퍼바이저를 AWS의 VMware Cloud로 재배치합니다.
- 유지(보관) - 소스 환경에 애플리케이션을 유지합니다. 대규모 리팩터링이 필요하고 해당 작업을 나중으로 연기하려는 애플리케이션과 비즈니스 차원에서 마이그레이션할 이유가 없어 유지하려는 레거시 애플리케이션이 여기에 포함될 수 있습니다.
- 사용 중지 - 소스 환경에서 더 이상 필요하지 않은 애플리케이션을 폐기하거나 제거합니다.

A

ABAC

속성 기반 액세스 제어를 참조하십시오.

추상화된 서비스

매니지드 서비스를 참조하십시오.

산

원자성, 일관성, 격리성, 내구성을 참조하십시오.

능동-능동 마이그레이션

양방향 복제 도구 또는 이중 쓰기 작업을 사용하여 소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되고, 두 데이터베이스 모두 마이그레이션 중 연결 애플리케이션의 트랜잭션을 처리하는 데이터베이스 마이그레이션 방법입니다. 이 방법은 일회성 전환이 필요한 대신 소규모의 제어된 배치로 마이그레이션을 지원합니다. 더 유연하지만 액티브-패시브 마이그레이션보다 더 많은 작업이 필요합니다.

능동-수동 마이그레이션

소스 데이터베이스와 대상 데이터베이스가 동기화된 상태로 유지되지만 소스 데이터베이스만 연결 애플리케이션의 트랜잭션을 처리하고 데이터는 대상 데이터베이스로 복제되는 데이터베이스 마이그레이션 방법입니다. 대상 데이터베이스는 마이그레이션 중 어떤 트랜잭션도 허용하지 않습니다.

집계 함수

행 그룹에서 연산을 수행하고 그룹에 대한 단일 반환값을 계산하는 SQL 함수입니다. 집계 함수의 예로는 및 등이 `SUM` 있습니다. `MAX`

AI

인공 지능을 참조하십시오.

AIOps

인공 지능 운영을 참조하십시오.

익명화

데이터세트에서 개인 정보를 영구적으로 삭제하는 프로세스입니다. 익명화는 개인 정보 보호에 도움이 될 수 있습니다. 익명화된 데이터는 더 이상 개인 데이터로 간주되지 않습니다.

안티 패턴

솔루션이 다른 솔루션보다 비생산적이거나 비효율적이거나 덜 효과적이어서 반복되는 문제에 자주 사용되는 솔루션입니다.

애플리케이션 제어

시스템을 멀웨어로부터 보호하기 위해 승인된 애플리케이션만 사용할 수 있는 보안 접근 방식입니다.

애플리케이션 포트폴리오

애플리케이션 구축 및 유지 관리 비용과 애플리케이션의 비즈니스 가치를 비롯하여 조직에서 사용하는 각 애플리케이션에 대한 세부 정보 모음입니다. 이 정보는 포트폴리오 검색 및 분석 프로세스의 핵심이며 마이그레이션, 현대화 및 최적화할 애플리케이션을 식별하고 우선순위를 정하는 데 도움이 됩니다.

인공 지능

컴퓨터 기술을 사용하여 학습, 문제 해결, 패턴 인식 등 일반적으로 인간과 관련된 인지 기능을 수행하는 것을 전문으로 하는 컴퓨터 과학 분야입니다. 자세한 내용은 What is Artificial Intelligence?를 참조하세요.

인공 지능 운영(AIOps)

기계 학습 기법을 사용하여 운영 문제를 해결하고, 운영 인시던트 및 사용자 개입을 줄이고, 서비스 품질을 높이는 프로세스입니다. AWS 마이그레이션 전략에서 AIOps가 사용되는 방법에 대한 자세한 내용은 운영 통합 가이드를 참조하세요.

비대칭 암호화

한 쌍의 키, 즉 암호화를 위한 퍼블릭 키와 복호화를 위한 프라이빗 키를 사용하는 암호화 알고리즘입니다. 퍼블릭 키는 복호화에 사용되지 않으므로 공유할 수 있지만 프라이빗 키에 대한 액세스는 엄격히 제한되어야 합니다.

원자성, 일관성, 격리성, 내구성(ACID)

오류, 정전 또는 기타 문제가 발생한 경우에도 데이터베이스의 데이터 유효성과 운영 신뢰성을 보장하는 소프트웨어 속성 세트입니다.

ABAC(속성 기반 액세스 제어)

부서, 직무, 팀 이름 등의 사용자 속성을 기반으로 세분화된 권한을 생성하는 방식입니다. 자세한 내용은 AWS Identity and Access Management(IAM) 설명서의 AWS용 ABAC란 무엇입니까?를 참조하세요.

신뢰할 수 있는 데이터 소스

가장 신뢰할 수 있는 정보 소스로 간주되는 기본 버전의 데이터를 저장하는 위치입니다. 익명화, 편집 또는 가명화와 같은 데이터 처리 또는 수정의 목적으로 신뢰할 수 있는 데이터 소스의 데이터를 다른 위치로 복사할 수 있습니다.

가용 영역

다른 가용 영역에 장애가 발생할 경우 분리되도록 설계된 AWS 리전 내의 개별적인 지점으로, 같은 리전 내의 다른 가용 영역에 비해 저렴하고 지연 시간이 짧은 네트워크 연결을 제공합니다.

AWS Cloud Adoption Framework(AWS CAF)

조직이 클라우드로 성공적으로 전환하기 위한 효율적이고 효과적인 계획을 개발하는 데 도움이 되는 AWS의 지침 및 모범 사례 프레임워크입니다. AWS CAF는 비즈니스, 사람, 거버넌스, 플랫폼, 보안 및 운영 관점이라는 6가지 중점 영역으로 지침을 구성합니다. 비즈니스, 사람 및 거버넌스 관점은 비즈니스 기술과 프로세스에 초점을 맞추고, 플랫폼, 보안 및 운영 관점은 전문 기술과 프로세스에 중점을 둡니다. 예를 들어, 사람 관점은 인사(HR), 직원 배치 기능 및 인력 관리를 담당하는 이해관계자를 대상으로 합니다. 이러한 관점에서 AWS CAF는 조직이 클라우드를 성공적으로 채택할 수 있도록 인력 개발, 교육 및 커뮤니케이션에 대한 지침을 제공합니다. 자세한 내용은 AWS CAF 웹 사이트와 AWS CAF 백서를 참조하세요.

AWS Workload Qualification Framework(AWS WQF)

데이터베이스 마이그레이션 워크로드를 평가하고, 마이그레이션 전략을 추천하고, 작업 추정치를 제공하는 도구입니다. AWS WQF는 AWS Schema Conversion Tool(AWS SCT)에 포함되어 있으며, 데이터베이스 스키마 및 코드 객체, 애플리케이션 코드, 종속성 및 성능 특성을 분석하고 평가 보고서를 제공합니다.

B

BCP

비즈니스 연속성 계획을 참조하십시오.

동작 그래프

리소스 동작과 시간 경과에 따른 상호 작용에 대한 통합된 대화형 뷰입니다. Amazon Detective에서 동작 그래프를 사용하여 실패한 로그온 시도, 의심스러운 API 호출 및 유사한 작업을 검사할 수 있습니다. 자세한 내용은 Detective 설명서의 Data in a behavior graph를 참조하세요.

빅 엔디안 시스템

가장 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하십시오.

바이너리 분류

바이너리 결과(가능한 두 클래스 중 하나)를 예측하는 프로세스입니다. 예를 들어, ML 모델이 '이 이메일이 스팸인가요, 스팸이 아닌가요?', '이 제품은 책인가요, 자동차인가요?' 등의 문제를 예측해야 할 수 있습니다.

블룸 필터

요소가 세트의 멤버인지 여부를 테스트하는 데 사용되는 메모리 효율성이 높은 확률론적 데이터 구조입니다.

브랜치

코드 리포지토리의 포함된 영역입니다. 리포지토리에 생성되는 첫 번째 브랜치가 기본 브랜치입니다. 기존 브랜치에서 새 브랜치를 생성한 다음 새 브랜치에서 기능을 개발하거나 버그를 수정할 수 있습니다. 기능을 구축하기 위해 생성하는 브랜치를 일반적으로 기능 브랜치라고 합니다. 기능을 출시할 준비가 되면 기능 브랜치를 기본 브랜치에 다시 병합합니다. 자세한 내용은 브랜치 정보(GitHub 문서) 를 참조하십시오.

브레이크 글래스 액세스

예외적인 상황에서 승인된 프로세스를 통해 사용자가 일반적으로 액세스 권한이 없는 데이터에 빠르게 액세스할 수 AWS 계정 있는 수단입니다. 자세한 내용은 Well-Architected AWS 지침의 브레이크 글래스 절차 구현 표시기를 참조하십시오.

브라운필드 전략

사용자 환경의 기존 인프라 시스템 아키텍처에 브라운필드 전략을 채택할 때는 현재 시스템 및 인프라의 제약 조건을 중심으로 아키텍처를 설계합니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 그린필드 전략을 혼합할 수 있습니다.

버퍼 캐시

가장 자주 액세스하는 데이터가 저장되는 메모리 영역입니다.

사업 역량

기업이 가치를 창출하기 위해 하는 일(예: 영업, 고객 서비스 또는 마케팅)입니다. 마이크로서비스 아키텍처 및 개발 결정은 비즈니스 역량에 따라 이루어질 수 있습니다. 자세한 내용은 백서의 컨테이너화된 마이크로서비스 실행AWS의 비즈니스 역량 중심의 구성화 섹션을 참조하세요.

비즈니스 연속성 계획(BCP)

대규모 마이그레이션과 같은 중단 이벤트가 운영에 미치는 잠재적 영향을 해결하고 비즈니스가 신속하게 운영을 재개할 수 있도록 지원하는 계획입니다.

C

CAF

클라우드 채택 프레임워크를 참조하십시오AWS.

CCoE

클라우드 센터 오브 엑설런스를 참조하십시오.

CDC

변경 데이터 캡처를 참조하십시오.

변경 데이터 캡처(CDC)

데이터베이스 테이블과 같은 데이터 소스의 변경 내용을 추적하고 변경 사항에 대한 메타데이터를 기록하는 프로세스입니다. 대상 시스템의 변경 내용을 감사하거나 복제하여 동기화를 유지하는 등의 다양한 용도로 CDC를 사용할 수 있습니다.

카오스 엔지니어링

시스템의 복원력을 테스트하기 위해 의도적으로 장애나 장애를 일으키는 이벤트를 발생시키는 행위 AWS Fault Injection Service(AWS FIS) 를 사용하여 AWS 워크로드에 스트레스를 주는 실험을 수행하고 응답을 평가할 수 있습니다.

CI/CD

지속적 통합 및 지속적 전달을 참조하십시오.

분류

예측을 생성하는 데 도움이 되는 분류 프로세스입니다. 분류 문제에 대한 ML 모델은 이산 값을 예측합니다. 이산 값은 항상 서로 다릅니다. 예를 들어, 모델이 이미지에 자동차가 있는지 여부를 평가해야 할 수 있습니다.

클라이언트측 암호화

데이터를 대상 AWS 서비스에서 수신하기 전에 로컬에서 암호화합니다.

클라우드 혁신 센터(CCoE)

클라우드 모범 사례 개발, 리소스 동원, 마이그레이션 타임라인 설정, 대규모 혁신을 통한 조직 선도 등 조직 전체에서 클라우드 채택 노력을 추진하는 다분야 팀입니다. 자세한 내용은 AWS 클라우드 엔터프라이즈 전략 블로그의 CCoE 게시물을 참조하세요.

클라우드 컴퓨팅

원격 데이터 스토리지와 IoT 디바이스 관리에 일반적으로 사용되는 클라우드 기술 클라우드 컴퓨팅은 일반적으로 엣지 컴퓨팅 기술과 연결됩니다.

클라우드 운영 모델

IT 조직에서 하나 이상의 클라우드 환경을 구축, 성숙화 및 최적화하는 데 사용되는 운영 모델입니다. 자세한 내용은 클라우드 운영 모델 구축을 참조하세요.

클라우드 채택 단계

조직이 AWS 클라우드로 마이그레이션할 때 일반적으로 거치는 4단계는 다음과 같습니다.

- 프로젝트 - 개념 증명 및 학습 목적으로 몇 가지 클라우드 관련 프로젝트 실행
- 기반 – 클라우드 채택 확장을 위한 기초 투자(예: 랜딩 존 생성, CCoE 정의, 운영 모델 구축)
- 마이그레이션 - 개별 애플리케이션 마이그레이션
- Re-invention - 제품 및 서비스 최적화와 클라우드 혁신

이러한 단계는 Stephen Orban이 AWS Cloud Enterprise Strategy Blog의 The Journey Toward Cloud-First & the Stages of Adoption 블로그 게시물에서 정의했습니다. AWS 마이그레이션 전략과 어떤 관련이 있는지에 대한 자세한 내용은 마이그레이션 준비 가이드를 참조하세요.

CMDB

구성 관리 데이터베이스를 참조하십시오.

코드 리포지토리

소스 코드와 설명서, 샘플, 스크립트 등의 기타 자산이 버전 관리 프로세스를 통해 저장되고 업데이트되는 위치입니다. 일반 클라우드 리포지토리에는 또는 이 포함됩니다 GitHub . AWS CodeCommit 코드의 각 버전을 브랜치라고 합니다. 마이크로서비스 구조에서 각 리포지토리는 단일 기능 전용입니다. 단일 CI/CD 파이프라인은 여러 리포지토리를 사용할 수 있습니다.

콜드 캐시

비어 있거나, 제대로 채워지지 않았거나, 오래되었거나 관련 없는 데이터를 포함하는 버퍼 캐시입니다. 주 메모리나 디스크에서 데이터베이스 인스턴스를 읽어야 하기 때문에 성능에 영향을 미치며, 이는 버퍼 캐시에서 읽는 것보다 느립니다.

콜드 데이터

거의 액세스되지 않고 일반적으로 과거 데이터인 데이터. 이런 종류의 데이터를 쿼리할 때는 일반적으로 느린 쿼리가 허용됩니다. 이 데이터를 성능이 낮고 비용이 저렴한 스토리지 계층 또는 클래스로 옮기면 비용을 절감할 수 있습니다.

컴퓨터 비전

기계가 이미지 속 사람, 장소, 사물을 인간 수준 이상의 정확도로 식별하는 데 사용하는 AI 분야입니다. 딥 러닝 모델로 구축되는 경우가 많으며 단일 이미지 또는 일련의 이미지에서 유용한 정보를 자동으로 추출, 분석, 분류 및 이해합니다.

구성 관리 데이터베이스(CMDB)

하드웨어 및 소프트웨어 구성 요소와 해당 구성을 포함하여 데이터베이스와 해당 IT 환경에 대한 정보를 저장하고 관리하는 리포지토리입니다. 일반적으로 마이그레이션의 포트폴리오 검색 및 분석 단계에서 CMDB의 데이터를 사용합니다.

규정 준수 팩

규정 준수 및 보안 검사를 사용자 지정하기 위해 조합할 수 있는 AWS Config 규칙 및 수정 작업 모음입니다. YAML 템플릿을 사용하여 AWS 계정 및 리전 또는 조직 전체에 단일 엔터티로 규정 준수 팩을 배포할 수 있습니다. 자세한 내용은 AWS Config 설명서의 Conformance packs를 참조하세요.

지속적 통합 및 지속적 전달(CI/CD)

소프트웨어 릴리스 프로세스의 소스, 빌드, 테스트, 스테이징 및 프로덕션 단계를 자동화하는 프로세스입니다. CI/CD는 일반적으로 파이프라인으로 설명됩니다. CI/CD를 통해 프로세스를 자동화하고, 생산성을 높이고, 코드 품질을 개선하고, 더 빠르게 제공할 수 있습니다. 자세한 내용은 지속적 전달의 이점을 참조하세요. CD는 지속적 배포를 의미하기도 합니다. 자세한 내용은 지속적 전달(Continuous Delivery)과 지속적인 개발을 참조하세요.

D

저장 데이터

스토리지에 있는 데이터와 같이 네트워크에 고정되어 있는 데이터입니다.

데이터 분류

중요도와 민감도를 기준으로 네트워크의 데이터를 식별하고 분류하는 프로세스입니다. 이 프로세스는 데이터에 대한 적절한 보호 및 보존 제어를 결정하는 데 도움이 되므로 사이버 보안 위험 관리

전략의 중요한 구성 요소입니다. 데이터 분류는 AWS Well-Architected Framework 보안 원칙의 구성 요소입니다. 자세한 내용은 데이터 분류를 참조하세요.

데이터 드리프트

프로덕션 데이터와 ML 모델 학습에 사용된 데이터 간의 상당한 차이 또는 시간 경과에 따른 입력 데이터의 의미 있는 변화. 데이터 드리프트는 ML 모델 예측의 전반적인 품질, 정확성 및 공정성을 저하시킬 수 있습니다.

전송 중 데이터

네트워크를 통과하고 있는 데이터입니다. 네트워크 리소스 사이를 이동 중인 데이터를 예로 들 수 있습니다.

데이터 최소화

꼭 필요한 데이터만 수집하고 처리하는 원칙입니다. AWS 클라우드에서 데이터 최소화를 실천하면 개인 정보 보호 위험, 비용 및 분석에 따른 탄소 발자국을 줄일 수 있습니다.

데이터 경계

신뢰할 수 있는 ID만 예상 네트워크에서 신뢰할 수 있는 리소스에 액세스하도록 하는 데 도움이 되는 AWS 환경 내 일련의 예방 가드레일입니다. 자세한 내용은 데이터 경계 구축을 참조하십시오. AWS

데이터 사전 처리

원시 데이터를 ML 모델이 쉽게 구문 분석할 수 있는 형식으로 변환하는 것입니다. 데이터를 사전 처리한다는 것은 특정 열이나 행을 제거하고 누락된 값, 일관성이 없는 값 또는 중복 값을 처리함을 의미할 수 있습니다.

데이터 출처

라이프사이클 전반에 걸쳐 데이터의 출처와 기록을 추적하는 프로세스(예: 데이터 생성, 전송, 저장 방법).

데이터 주체

데이터를 수집 및 처리하는 개인입니다.

데이터 웨어하우스

분석과 같은 비즈니스 인텔리전스를 지원하는 데이터 관리 시스템. 데이터 웨어하우스에는 일반적으로 대량의 과거 데이터가 포함되며 일반적으로 쿼리 및 분석에 사용됩니다.

데이터 정의 언어(DDL)

데이터베이스에서 테이블 및 객체의 구조를 만들거나 수정하기 위한 명령문 또는 명령입니다.

데이터베이스 조작 언어(DML)

데이터베이스에서 정보를 수정(삽입, 업데이트 및 삭제)하기 위한 명령문 또는 명령입니다.

DDL

데이터베이스 정의 언어를 참조하십시오.

딥 앙상블

예측을 위해 여러 딥 러닝 모델을 결합하는 것입니다. 딥 앙상블을 사용하여 더 정확한 예측을 얻거나 예측의 불확실성을 추정할 수 있습니다.

딥 러닝

여러 계층의 인공 신경망을 사용하여 입력 데이터와 관심 대상 변수 간의 매핑을 식별하는 ML 하위 분야입니다.

defense-in-depth

네트워크와 그 안의 데이터 기밀성, 무결성 및 가용성을 보호하기 위해 컴퓨터 네트워크 전체에 일련의 보안 메커니즘과 제어를 신중하게 계층화하는 정보 보안 접근 방식입니다. AWS에서 이 전략을 채택하면 AWS Organizations 구조의 다양한 계층에 여러 제어 기능을 추가하여 리소스를 보호할 수 있습니다. 예를 들어, defense-in-depth 접근 방식에는 다단계 인증, 네트워크 분할 및 암호화를 결합할 수 있습니다.

위임된 관리자

AWS Organizations에서 호환되는 서비스는 AWS 멤버 계정을 등록하여 조직의 계정을 관리하고 해당 서비스에 대한 권한을 관리할 수 있습니다. 이러한 계정을 해당 서비스의 위임된 관리자라고 합니다. 자세한 내용과 호환되는 서비스 목록은 AWS Organizations 설명서의 AWS Organizations와 함께 사용할 수 있는 AWS 서비스를 참조하세요.

배포

대상 환경에서 애플리케이션, 새 기능 또는 코드 수정 사항을 사용할 수 있도록 하는 프로세스입니다. 배포에는 코드 베이스의 변경 사항을 구현한 다음 애플리케이션 환경에서 해당 코드베이스를 구축하고 실행하는 작업이 포함됩니다.

개발 환경

환경을 참조하십시오.

탐지 제어

이벤트 발생 후 탐지, 기록 및 알림을 수행하도록 설계된 보안 제어입니다. 이러한 제어는 기존의 예방적 제어를 우회한 보안 이벤트를 알리는 2차 방어선입니다. 자세한 내용은 Implementing security controls on AWS의 Detective controls를 참조하세요.

개발 가치 흐름 매핑 (DVSM)

소프트웨어 개발 라이프사이클에서 속도와 품질에 부정적인 영향을 미치는 제약 조건을 식별하고 우선 순위를 지정하는 데 사용되는 프로세스입니다. DVSM은 원래 린 제조 방식을 위해 설계된 가치 흐름 매핑 프로세스를 확장합니다. 소프트웨어 개발 프로세스를 통해 가치를 창출하고 이동하는 데 필요한 단계와 팀에 중점을 둡니다.

디지털 트윈

건물, 공장, 산업 장비 또는 생산 라인과 같은 실제 시스템을 가상으로 표현한 것입니다. 디지털 트윈은 예측 유지 보수, 원격 모니터링, 생산 최적화를 지원합니다.

치수 표

스타 스키마에서 팩트 테이블의 양적 데이터에 대한 데이터 속성을 포함하는 작은 테이블입니다. 차원 테이블 속성은 일반적으로 텍스트처럼 동작하는 텍스트 필드 또는 불연속형 숫자입니다. 이러한 속성은 일반적으로 쿼리 제한, 필터링 및 결과 집합 레이블 지정에 사용됩니다.

재해

워크로드 또는 시스템이 기본 배포 위치에서 비즈니스 목표를 달성하지 못하게 방해하는 이벤트입니다. 이러한 이벤트는 자연재해, 기술적 오류, 의도하지 않은 구성 오류 또는 멀웨어 공격과 같은 사람의 행동으로 인한 결과일 수 있습니다.

재해 복구(DR)

재해로 인한 다운타임과 데이터 손실을 최소화하기 위해 사용하는 전략과 프로세스입니다. 자세한 내용은 AWS Well-Architected Framework의 AWS 기반 워크로로드의 재해 복구: 클라우드에서의 재해 복구를 참조하세요.

DML

데이터베이스 조작 언어를 참조하십시오.

도메인 기반 설계

구성 요소를 각 구성 요소가 제공하는 진화하는 도메인 또는 핵심 비즈니스 목표에 연결하여 복잡한 소프트웨어 시스템을 개발하는 접근 방식입니다. 이 개념은 에릭 에반스에 의해 그의 저서인 도

메인 기반 디자인: 소프트웨어 중심의 복잡성 해결(Boston: Addison-Wesley Professional, 2003)에서 소개되었습니다. Strangler Fig 패턴과 함께 도메인 기반 설계를 사용하는 방법에 대한 자세한 내용은 컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET(ASMX) 웹 서비스를 점진적으로 현대화하는 방법을 참조하세요.

DR

재해 복구를 참조하십시오.

드리프트 감지

기준 구성으로부터의 편차 추적 예를 들어 시스템 리소스의 편차를 감지하는 AWS CloudFormation 데 사용하거나 거버넌스 요구 사항 준수에 영향을 미칠 수 있는 착륙 지대의 변경을 탐지하는 AWS Control Tower 데 사용할 수 있습니다.

DVSM

개발 가치 흐름 매핑을 참조하십시오.

E

EDA

탐색적 데이터 분석을 참조하십시오.

엣지 컴퓨팅

IoT 네트워크의 엣지에서 스마트 디바이스의 컴퓨팅 성능을 개선하는 기술 클라우드 컴퓨팅과 비교할 때 엣지 컴퓨팅은 통신 대기 시간을 줄이고 응답 시간을 개선할 수 있습니다.

암호화

사람이 읽을 수 있는 일반 텍스트 데이터를 암호문으로 변환하는 컴퓨팅 프로세스입니다.

암호화 키

암호화 알고리즘에 의해 생성되는 무작위 비트의 암호화 문자열입니다. 키의 길이는 다양할 수 있으며 각 키는 예측할 수 없고 고유하게 설계되었습니다.

엔디안

컴퓨터 메모리에 바이트가 저장되는 순서입니다. 빅 엔디안 시스템은 가장 중요한 바이트를 먼저 저장합니다. 리틀 엔디안 시스템은 가장 덜 중요한 바이트를 먼저 저장합니다.

엔드포인트

서비스 엔드포인트를 참조하십시오.

엔드포인트 서비스

Virtual Private Cloud(VPC)에서 호스팅하여 다른 사용자와 공유할 수 있는 서비스입니다. AWS PrivateLink를 사용하여 엔드포인트 서비스를 생성하고 다른 AWS 계정 또는 AWS Identity and Access Management(IAM) 보안 주체에게 권한을 부여할 수 있습니다. 이러한 계정 또는 보안 주체는 인터페이스 VPC 엔드포인트를 생성하여 엔드포인트 서비스에 비공개로 연결할 수 있습니다. 자세한 내용은 Amazon Virtual Private Cloud(VPC) 설명서의 엔드포인트 서비스 생성을 참조하세요.

봉투 암호화

암호화 키를 다른 암호화 키로 암호화하는 프로세스입니다. 자세한 내용은 AWS Key Management Service(AWS KMS) 설명서의 Envelope encryption을 참조하세요.

환경

실행 중인 애플리케이션의 인스턴스입니다. 다음은 클라우드 컴퓨팅의 일반적인 환경 유형입니다.

- 개발 환경 - 애플리케이션 유지 관리를 담당하는 핵심 팀만 사용할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. 개발 환경은 변경 사항을 상위 환경으로 승격하기 전에 테스트하는 데 사용됩니다. 이러한 유형의 환경을 테스트 환경이라고도 합니다.
- 하위 환경 - 초기 빌드 및 테스트에 사용되는 환경을 비롯한 애플리케이션의 모든 개발 환경입니다.
- 프로덕션 환경 - 최종 사용자가 액세스할 수 있는 실행 중인 애플리케이션의 인스턴스입니다. CI/CD 파이프라인에서 프로덕션 환경이 마지막 배포 환경입니다.
- 상위 환경 - 핵심 개발 팀 이외의 사용자가 액세스할 수 있는 모든 환경입니다. 프로덕션 환경, 프로덕션 이전 환경 및 사용자 수용 테스트를 위한 환경이 여기에 포함될 수 있습니다.

에픽

애자일 방법론에서 작업을 구성하고 우선순위를 정하는 데 도움이 되는 기능적 범주입니다. 에픽은 요구 사항 및 구현 작업에 대한 개괄적인 설명을 제공합니다. 예를 들어, AWS CAF 보안 에픽에는 ID 및 액세스 관리, 탐지 제어, 인프라 보안, 데이터 보호 및 인시던트 대응 등이 포함됩니다. AWS 마이그레이션 전략의 에픽에 대한 자세한 내용은 프로그램 구현 가이드를 참조하세요.

탐색 데이터 분석(EDA)

데이터 세트를 분석하여 주요 특성을 파악하는 프로세스입니다. 데이터를 수집 또는 집계한 다음 초기 조사를 수행하여 패턴을 찾고, 이상을 탐지하고, 가정을 확인합니다. EDA는 요약 통계를 계산하고 데이터 시각화를 생성하여 수행됩니다.

F

팩트 테이블

[스타 스키마의](#) 중앙 테이블. 비즈니스 운영에 대한 정량적 데이터를 저장합니다. 일반적으로 팩트 테이블에는 측정값이 포함된 열과 차원 테이블의 외부 키가 포함된 열 등 두 가지 유형의 열이 포함됩니다.

빨리 실패하세요

빈번하고 점진적인 테스트를 통해 개발 라이프사이클을 단축하는 철학. 이는 애자일 접근 방식의 중요한 부분입니다.

장애 격리 경계

장애 영향을 제한하고 워크로드의 복원력을 개선하는 데 도움이 되는 가용 영역AWS 리전, 컨트롤 플레인 또는 데이터 플레인과 같은 경계 AWS 클라우드 자세한 내용은 [AWS장애 격리](#) 경계를 참조하십시오.

기능 브랜치

[브랜치를](#) 참조하십시오.

기능

예측에 사용하는 입력 데이터입니다. 예를 들어, 제조 환경에서 기능은 제조 라인에서 주기적으로 캡처되는 이미지일 수 있습니다.

기능 중요도

모델의 예측에 특성이 얼마나 중요한지를 나타냅니다. 이는 일반적으로 SHAP(Shapley Additive Descriptions) 및 통합 그레디언트와 같은 다양한 기법을 통해 계산할 수 있는 수치 점수로 표현됩니다. 자세한 내용은 [다음을 AWS 사용한 기계 학습 모델 해석 가능성을](#) 참조하십시오.

기능 변환

추가 소스로 데이터를 보강하거나, 값을 조정하거나, 단일 데이터 필드에서 여러 정보 세트를 추출하는 등 ML 프로세스를 위해 데이터를 최적화하는 것입니다. 이를 통해 ML 모델이 데이터를 활용할 수 있습니다. 예를 들어, 날짜 '2021-05-27 00:15:37'을 '2021년', '5월', '목', '15일'로 분류하면 학습 알고리즘이 다양한 데이터 구성 요소와 관련된 미묘한 패턴을 학습하는 데 도움이 됩니다.

FGAC

[세분화된 액세스 제어를](#) 참조하십시오.

세분화된 액세스 제어(FGAC)

여러 조건을 사용하여 액세스 요청을 허용하거나 거부합니다.

플래시컷 마이그레이션

단계별 접근 방식 대신 변경 데이터 캡처를 통한 지속적인 데이터 복제를 통해 최단 시간에 데이터를 마이그레이션하는 데이터베이스 마이그레이션 방법입니다. 목표는 가동 중지 시간을 최소화하는 것입니다.

G

지리적 차단

지리적 제한을 참조하십시오.

지리적 제한(지리적 차단)

CloudFrontAmazon에서는 특정 국가의 사용자가 콘텐츠 배포에 액세스하지 못하도록 하는 옵션을 제공합니다. 허용 목록 또는 차단 목록을 사용하여 승인된 국가와 차단된 국가를 지정할 수 있습니다. 자세한 내용은 설명서의 콘텐츠의 지리적 배포 제한을 참조하십시오. CloudFront

Gitflow 워크플로

하위 환경과 상위 환경이 소스 코드 리포지토리의 서로 다른 브랜치를 사용하는 방식입니다. Gitflow 워크플로는 레거시로 간주되며 트렁크 기반 워크플로는 현대적이고 선호되는 접근 방식입니다.

브라운필드 전략

새로운 환경에서 기존 인프라의 부재 시스템 아키텍처에 대한 그린필드 전략을 채택할 때 브라운필드라고도 하는 기존 인프라와의 호환성 제한 없이 모든 새로운 기술을 선택할 수 있습니다. 기존 인프라를 확장하는 경우 브라운필드 전략과 그린필드 전략을 혼합할 수 있습니다.

가드레일

조직 단위(OU) 전체에서 리소스, 정책 및 규정 준수를 관리하는 데 도움이 되는 중요 규칙입니다. 예방 가드레일은 규정 준수 표준에 부합하도록 정책을 시행하며, 서비스 제어 정책과 IAM 권한 경계를 사용하여 구현됩니다. 탐지 가드레일은 정책 위반 및 규정 준수 문제를 감지하고 해결을 위한 알림을 생성하며, 이들은, Amazon AWS Config AWS Security Hub GuardDutyAWS Trusted Advisor, Amazon Inspector 및 사용자 지정 AWS Lambda 검사를 사용하여 구현됩니다.

H

하

고가용성을 확인하세요.

이기종 데이터베이스 마이그레이션

다른 데이터베이스 엔진을 사용하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Oracle에서 Amazon Aurora로) 이기종 마이그레이션은 일반적으로 리아키텍트 작업의 일부이며 스키마를 변환하는 것은 복잡한 작업일 수 있습니다. AWS는 스키마 변환에 도움이 되는 AWS SCT를 제공합니다.

높은 가용성(HA)

문제나 재해 발생 시 개입 없이 지속적으로 운영할 수 있는 워크로드의 능력. HA 시스템은 자동으로 장애 조치되고, 지속적으로 고품질 성능을 제공하고, 성능에 미치는 영향을 최소화하면서 다양한 부하와 장애를 처리하도록 설계되었습니다.

히스토리언 현대화

제조 산업의 요구 사항을 더 잘 충족하도록 운영 기술(OT) 시스템을 현대화하고 업그레이드하는 데 사용되는 접근 방식입니다. 히스토리언은 공장의 다양한 출처에서 데이터를 수집하고 저장하는 데 사용되는 일종의 데이터베이스입니다.

동종 데이터베이스 마이그레이션

동일한 데이터베이스 엔진을 공유하는 대상 데이터베이스로 소스 데이터베이스 마이그레이션(예: Microsoft SQL Server에서 Amazon RDS for SQL Server로) 동종 마이그레이션은 일반적으로 리호스팅 또는 리플랫포밍 작업의 일부입니다. 네이티브 데이터베이스 유틸리티를 사용하여 스키마를 마이그레이션할 수 있습니다.

핫 데이터

자주 액세스하는 데이터(예: 실시간 데이터 또는 최근 번역 데이터). 일반적으로 이 데이터에는 빠른 쿼리 응답을 제공하기 위한 고성능 스토리지 계층 또는 클래스가 필요합니다.

핫픽스

프로덕션 환경의 중요한 문제를 해결하기 위한 긴급 수정입니다. 긴급성 때문에 핫픽스는 일반적으로 일반적인 DevOps 릴리스 워크플로 외부에서 만들어집니다.

하이퍼케어 기간

전환 직후 마이그레이션 팀이 문제를 해결하기 위해 클라우드에서 마이그레이션된 애플리케이션을 관리하고 모니터링하는 기간입니다. 일반적으로 이 기간은 1~4일입니다. 하이퍼케어 기간이 끝나면 마이그레이션 팀은 일반적으로 애플리케이션에 대한 책임을 클라우드 운영 팀에 넘깁니다.

I

IaC

인프라를 코드로 보세요.

ID 기반 정책

하나 이상의 IAM 보안 주체에 연결된 정책으로, AWS 클라우드 클라우드 환경 내에서 IAM 보안 주체의 권한을 정의합니다.

유휴 애플리케이션

90일 동안 평균 CPU 및 메모리 사용량이 5~20%인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하거나 온프레미스에 유지하는 것이 일반적입니다.

IIoT

산업용 사물 인터넷을 참조하십시오.

불변의 인프라

기존 인프라를 업데이트, 패치 또는 수정하는 대신 프로덕션 워크로드용 새 인프라를 배포하는 모델입니다. 변경 불가능한 인프라는 기본적으로 변경 가능한 인프라보다 더 일관되고 안정적이며 예측 가능합니다. 자세한 내용은 Well-Architected AWS 프레임워크의 변경 불가능한 인프라를 사용한 배포 모범 사례를 참조하십시오.

인바운드(수신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 외부의 네트워크 연결을 수락, 검사 및 라우팅하는 VPC입니다. AWS Security Reference Architecture에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

증분 마이그레이션

한 번에 전체 전환을 수행하는 대신 애플리케이션을 조금씩 마이그레이션하는 전환 전략입니다. 예를 들어, 처음에는 소수의 마이크로서비스나 사용자만 새 시스템으로 이동할 수 있습니다. 모든 것

이 제대로 작동하는지 확인한 후에는 레거시 시스템을 폐기할 수 있을 때까지 추가 마이크로서비스 또는 사용자를 점진적으로 이동할 수 있습니다. 이 전략을 사용하면 대규모 마이그레이션과 관련된 위험을 줄일 수 있습니다.

인프라

애플리케이션의 환경 내에 포함된 모든 리소스와 자산입니다.

코드형 인프라(IaC)

구성 파일 세트를 통해 애플리케이션의 인프라를 프로비저닝하고 관리하는 프로세스입니다. IaC는 새로운 환경의 반복 가능성, 신뢰성 및 일관성을 위해 인프라 관리를 중앙 집중화하고, 리소스를 표준화하고, 빠르게 확장할 수 있도록 설계되었습니다.

산업용 사물 인터넷(IIoT)

제조, 에너지, 자동차, 의료, 생명과학, 농업 등의 산업 부문에서 인터넷에 연결된 센서 및 디바이스의 사용 자세한 내용은 산업용 사물 인터넷(IoT) 디지털 트랜스포메이션 전략 구축을 참조하세요.

검사 VPC

AWS 다중 계정 아키텍처에서 동일한 또는 다른 AWS 리전에 있는 VPC, 인터넷 및 온프레미스 네트워크 간의 네트워크 트래픽 검사를 관리하는 중앙 집중식 VPC입니다. AWS Security Reference Architecture에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

사물 인터넷(IoT)

인터넷이나 로컬 통신 네트워크를 통해 다른 디바이스 및 시스템과 통신하는 센서 또는 프로세서가 내장된 연결된 물리적 객체의 네트워크 자세한 내용은 IoT란?을 참조하세요.

해석력

모델의 예측이 입력에 따라 어떻게 달라지는지를 사람이 이해할 수 있는 정도를 설명하는 기계 학습 모델의 특성입니다. 자세한 내용은 Machine learning model interpretability with AWS를 참조하세요.

IoT

사물 인터넷을 참조하십시오.

IT 정보 라이브러리(TIL)

IT 서비스를 제공하고 이러한 서비스를 비즈니스 요구 사항에 맞게 조정하기 위한 일련의 모범 사례 ITIL은 ITSM의 기반을 제공합니다.

IT 서비스 관리(TSM)

조직의 IT 서비스 설계, 구현, 관리 및 지원과 관련된 활동 클라우드 운영을 ITSM 도구와 통합하는 방법에 대한 자세한 내용은 운영 통합 가이드를 참조하세요.

ITIL

IT 정보 라이브러리를 참조하십시오.

ITSM

IT 서비스 관리를 참조하십시오.

L

레이블 기반 액세스 제어(LBAC)

사용자 및 데이터 자체에 각각 보안 레이블 값을 명시적으로 할당하는 필수 액세스 제어(MAC)를 구현한 것입니다. 사용자 보안 레이블과 데이터 보안 레이블 간의 교차 부분에 따라 사용자가 볼 수 있는 행과 열이 결정됩니다.

랜딩 존

랜딩 존은 확장성과 안전성을 갖춘 잘 설계된 다중 계정 AWS 환경입니다. 조직은 여기에서부터 보안 및 인프라 환경에 대한 확신을 가지고 워크로드와 애플리케이션을 신속하게 시작하고 배포할 수 있습니다. 랜딩 존에 대한 자세한 내용은 Setting up a secure and scalable multi-account AWS environment를 참조하세요.

대규모 마이그레이션

300대 이상의 서버 마이그레이션입니다.

LBAC

레이블 기반 액세스 제어를 참조하십시오.

최소 권한

작업을 수행하는 데 필요한 최소 권한을 부여하는 보안 모범 사례입니다. 자세한 내용은 IAM 설명서의 최소 권한 적용을 참조하세요.

리프트 앤드 시프트

7 R을 참조하십시오.

리틀 엔디안 시스템

가장 덜 중요한 바이트를 먼저 저장하는 시스템입니다. 엔디안도 참조하십시오.

하위 환경

환경 참조.

M

기계 학습(ML)

패턴 인식 및 학습에 알고리즘과 기법을 사용하는 인공 지능의 한 유형입니다. ML은 사물 인터넷(IoT) 데이터와 같은 기록된 데이터를 분석하고 학습하여 패턴을 기반으로 통계 모델을 생성합니다. 자세한 내용은 기계 학습을 참조하세요.

기본 브랜치

브랜치 참조.

매니지드 서비스

AWS 서비스인프라 계층, 운영 체제 및 플랫폼을 AWS 운영하며 사용자는 엔드포인트에 액세스하여 데이터를 저장하고 검색합니다. 관리형 서비스의 예로는 아마존 심플 스토리지 서비스 (Amazon S3) 와 아마존 DynamoDB가 있습니다. 이러한 서비스를 추상화된 서비스라고도 합니다.

MAP

Migration Acceleration 프로그램을 참조하십시오.

기구

도구를 만들고 도구 채택을 유도한 다음 결과를 검토하여 조정하는 전체 프로세스입니다. 메커니즘은 작동하면서 자체적으로 강화되고 개선되는 사이클입니다. 자세한 내용은 AWS Well-Architected 프레임워크에서의 메커니즘 구축을 참조하십시오.

멤버 계정

AWS Organizations의 조직에 속하는 관리 계정 이외의 모든 AWS 계정입니다. 하나의 계정은 한 번에 하나의 조직 멤버만 될 수 있습니다.

마이크로서비스

잘 정의된 API를 통해 통신하고 일반적으로 소규모 자체 팀이 소유하는 소규모 독립 서비스입니다. 예를 들어, 보험 시스템에는 영업, 마케팅 등의 비즈니스 역량이나 구매, 청구, 분석 등의 하위 영역

에 매핑되는 마이크로 서비스가 포함될 수 있습니다. 마이크로서비스의 이점으로 민첩성, 유연한 확장, 손쉬운 배포, 재사용 가능한 코드, 복원력 등이 있습니다. 자세한 내용은 AWS 서버리스 서비스를 사용하여 마이크로서비스 통합을 참조하세요.

마이크로서비스 아키텍처

각 애플리케이션 프로세스를 마이크로서비스로 실행하는 독립 구성 요소를 사용하여 애플리케이션을 구축하는 접근 방식입니다. 이러한 마이크로서비스는 경량 API를 사용하여 잘 정의된 인터페이스를 통해 통신합니다. 애플리케이션의 특정 기능에 대한 수요에 맞게 이 아키텍처의 각 마이크로서비스를 업데이트, 배포 및 조정할 수 있습니다. 자세한 내용은 AWS에서 마이크로서비스 구현을 참조하세요.

Migration Acceleration Program(MAP)

조직이 클라우드로 전환하기 위한 강력한 운영 기반을 구축하고 마이그레이션 초기 비용을 상쇄할 수 있도록 컨설팅 지원, 교육 및 서비스를 제공하는 AWS 프로그램입니다. MAP에는 레거시 마이그레이션을 체계적인 방식으로 실행하기 위한 마이그레이션 방법론과 일반적인 마이그레이션 시나리오를 자동화하고 가속화하는 도구 세트가 포함되어 있습니다.

대규모 마이그레이션

애플리케이션 포트폴리오의 대다수를 웨이브를 통해 클라우드로 이동하는 프로세스로, 각 웨이브에서 더 많은 애플리케이션이 더 빠른 속도로 이동합니다. 이 단계에서는 이전 단계에서 배운 모범 사례와 교훈을 사용하여 팀, 도구 및 프로세스의 마이그레이션 팩토리를 구현하여 자동화 및 민첩한 제공을 통해 워크로드 마이그레이션을 간소화합니다. 이것은 AWS 마이그레이션 전략의 세 번째 단계입니다.

마이그레이션 팩토리

자동화되고 민첩한 접근 방식을 통해 워크로드 마이그레이션을 간소화하는 다기능 팀입니다. 마이그레이션 팩토리 팀에는 일반적으로 운영, 비즈니스 분석가 및 소유자, 마이그레이션 엔지니어, 개발자, 스프린트에서 일하는 DevOps 전문가가 포함됩니다. 엔터프라이즈 애플리케이션 포트폴리오의 20~50%는 공장 접근 방식으로 최적화할 수 있는 반복되는 패턴으로 구성되어 있습니다. 자세한 내용은 이 콘텐츠 세트의 클라우드 마이그레이션 팩토리 가이드와 마이그레이션 팩토리에 대한 설명을 참조하세요.

마이그레이션 메타데이터

마이그레이션을 완료하는 데 필요한 애플리케이션 및 서버에 대한 정보 각 마이그레이션 패턴에는 서로 다른 마이그레이션 메타데이터 세트가 필요합니다. 마이그레이션 메타데이터의 예로는 대상 서브넷, 보안 그룹 및 AWS 계정이 있습니다.

마이그레이션 패턴

사용되는 마이그레이션 전략, 마이그레이션 대상, 마이그레이션 애플리케이션 또는 서비스를 자세히 설명하는 반복 가능한 마이그레이션 작업입니다. 예를 들어, AWS Application Migration Service를 사용하여 Amazon EC2로 마이그레이션을 리호스팅합니다.

Migration Portfolio Assessment(MPA)

AWS 클라우드로 마이그레이션의 비즈니스 사례를 검증하기 위한 정보를 제공하는 온라인 도구입니다. MPA는 상세한 포트폴리오 평가(서버 적정 규모 조정, 가격 책정, TCO 비교, 마이그레이션 비용 분석)와 마이그레이션 계획(애플리케이션 데이터 분석 및 데이터 수집, 애플리케이션 그룹화, 마이그레이션 우선순위 지정, 웨이브 계획)을 제공합니다. MPA 도구(로그인 필요)는 모든 AWS 컨설턴트와 APN 파트너 컨설턴트에게 무료로 제공됩니다.

마이그레이션 준비 상태 평가(MRA)

AWS CAF를 사용하여 조직의 클라우드 준비 상태에 대한 인사이트를 얻고, 장단점을 파악하고, 파악된 격차를 해소하기 위한 실행 계획을 수립하는 프로세스입니다. 자세한 내용은 마이그레이션 준비 가이드를 참조하세요. MRA는 AWS 마이그레이션 전략의 첫 번째 단계입니다.

마이그레이션 전략

워크로드를 AWS 클라우드로 마이그레이션하는 데 사용되는 접근 방식입니다. 자세한 내용은 이 용어집의 7R 항목을 참조하고 대규모 마이그레이션을 가속화하기 위한 조직 동원을 참조하십시오.

ML

기계 학습을 참조하십시오.

MPA

마이그레이션 포트폴리오 평가를 참조하십시오.

현대화

비용을 절감하고 효율성을 높이고 혁신을 활용하기 위해 구식(레거시 또는 모놀리식) 애플리케이션과 해당 인프라를 클라우드의 민첩하고 탄력적이고 가용성이 높은 시스템으로 전환하는 것입니다. 자세한 내용은 AWS 클라우드에서 애플리케이션 현대화 전략을 참조하세요.

현대화 준비 상태 평가

조직 애플리케이션의 현대화 준비 상태를 파악하고, 이점, 위험 및 종속성을 식별하고, 조직이 해당 애플리케이션의 향후 상태를 얼마나 잘 지원할 수 있는지를 확인하는 데 도움이 되는 평가입니다. 평가 결과는 대상 아키텍처의 청사진, 현대화 프로세스의 개발 단계와 마일스톤을 자세히 설명하는

로드맵 및 파악된 격차를 해소하기 위한 실행 계획입니다. 자세한 내용은 AWS 클라우드에서 애플리케이션의 현대화 준비 상태 평가를 참조하세요.

모놀리식 애플리케이션(모놀리식 유형)

긴밀하게 연결된 프로세스를 사용하여 단일 서비스로 실행되는 애플리케이션입니다. 모놀리식 애플리케이션에는 몇 가지 단점이 있습니다. 한 애플리케이션 기능에 대한 수요가 급증하면 전체 아키텍처 규모를 조정해야 합니다. 코드 베이스가 커지면 모놀리식 애플리케이션의 기능을 추가하거나 개선하는 것도 더 복잡해집니다. 이러한 문제를 해결하기 위해 마이크로서비스 아키텍처를 사용할 수 있습니다. 자세한 내용은 마이크로서비스로 모놀리식 유형 분해를 참조하세요.

멀티클래스 분류

여러 클래스에 대한 예측(2개 이상의 결과 중 하나 예측)을 생성하는 데 도움이 되는 프로세스입니다. 예를 들어, ML 모델이 '이 제품은 책인가요, 자동차인가요, 휴대폰인가요?' 또는 '이 고객이 가장 관심을 갖는 제품 범주는 무엇인가요?'라고 물을 수 있습니다.

변경 가능한 인프라

프로덕션 워크로드를 위해 기존 인프라를 업데이트하고 수정하는 모델입니다. 일관성, 안정성 및 예측 가능성을 개선하기 위해 AWS Well-Architected Framework는 변경 불가능한 인프라를 모범 사례로 사용할 것을 권장합니다.

O

OAC

원본 액세스 제어를 참조하십시오.

좋아요

원본 액세스 ID를 참조하십시오.

OCM

조직 변경 관리를 참조하십시오.

오프라인 마이그레이션

마이그레이션 프로세스 중 소스 워크로드가 중단되는 마이그레이션 방법입니다. 이 방법은 가동 중지 증가를 수반하며 일반적으로 작고 중요하지 않은 워크로드에 사용됩니다.

OI

운영 통합을 참조하십시오.

안녕하세요.

운영 수준 계약을 참조하십시오.

온라인 마이그레이션

소스 워크로드를 오프라인 상태로 전환하지 않고 대상 시스템에 복사하는 마이그레이션 방법입니다. 워크로드에 연결된 애플리케이션은 마이그레이션 중에도 계속 작동할 수 있습니다. 이 방법은 가동 중지 차단 또는 최소화를 수반하며 일반적으로 중요한 프로덕션 워크로드에 사용됩니다.

운영 수준 협약(OLA)

서비스 수준에 관한 계약(SLA)을 지원하기 위해 직무 IT 그룹이 서로에게 제공하기로 약속한 내용을 명확히 하는 계약입니다.

운영 준비 상태 검토 (ORR)

인시던트 및 발생 가능한 실패의 범위를 이해, 평가, 예방 또는 줄이는 데 도움이 되는 질문 및 관련 모범 사례로 구성된 체크리스트입니다. 자세한 내용은 Well-Architected AWS 프레임워크의 운영 준비 상태 검토 (ORR) 를 참조하십시오.

운영 통합(OI)

클라우드에서 운영을 현대화하는 프로세스로 준비 계획, 자동화 및 통합을 수반합니다. 자세한 내용은 운영 통합 가이드를 참조하세요.

조직 트레일

AWS Organizations의 조직 내 모든 AWS 계정에 대한 모든 이벤트를 로깅하기 위해 AWS CloudTrail에서 생성하는 트레일입니다. 이 트레일은 조직에 속한 각 AWS 계정에 생성되고 각 계정의 활동을 추적합니다. 자세한 내용은 설명서의 조직을 위한 트레일 만들기를 참조하십시오. CloudTrail

조직 변경 관리(OCM)

사람, 문화 및 리더십 관점에서 중대하고 파괴적인 비즈니스 혁신을 관리하기 위한 프레임워크입니다. OCM은 변화 채택을 가속화하고, 과도기적 문제를 해결하고, 문화 및 조직적 변화를 주도함으로써 조직이 새로운 시스템 및 전략을 준비하고 전환할 수 있도록 지원합니다. AWS 마이그레이션 전략에서는 클라우드 채택 프로젝트에 필요한 변화 속도 때문에 이 프레임워크를 인력 가속화라고 합니다. 자세한 내용은 사용 가이드를 참조하세요.

오리진 액세스 제어(OAC)

CloudFront에서는 Amazon Simple Storage Service (Amazon S3) 콘텐츠의 보안을 위해 액세스를 제한하는 향상된 옵션을 제공합니다. OAC는 모든 AWS 리전의 모든 S3 버킷, AWS KMS(SSE-KMS)를 사용한 서버측 암호화, S3 버킷에 대한 동적 PUT 및 DELETE 요청을 지원합니다.

오리진 액세스 ID(OAI)

CloudFront에서는 Amazon S3 콘텐츠 보안을 위해 액세스를 제한하는 옵션입니다. OAI를 사용하면 Amazon S3가 인증할 수 있는 보안 주체를 CloudFront 생성합니다. 인증된 보안 주체는 특정 배포를 통해서만 S3 버킷의 콘텐츠에 액세스할 수 있습니다. CloudFront 더 세분화되고 향상된 액세스 제어를 제공하는 OAC도 참조하세요.

또는

운영 준비 상태 검토를 참조하십시오.

아웃바운드(송신) VPC

AWS 다중 계정 아키텍처에서 애플리케이션 내에서 시작되는 네트워크 연결을 처리하는 VPC입니다. AWS Security Reference Architecture에서는 애플리케이션과 더 넓은 인터넷 간의 양방향 인터페이스를 보호하기 위해 인바운드, 아웃바운드 및 검사 VPC로 네트워크 계정을 설정할 것을 권장합니다.

P

권한 경계

사용자나 역할이 가질 수 있는 최대 권한을 설정하기 위해 IAM 보안 주체에 연결되는 IAM 관리 정책입니다. 자세한 내용은 IAM 설명서의 권한 경계를 참조하세요.

개인 식별 정보(PII)

직접 보거나 다른 관련 데이터와 함께 짝을 지을 때 개인의 신원을 합리적으로 추론하는 데 사용할 수 있는 정보입니다. PII의 예로는 이름, 주소, 연락처 정보 등이 있습니다.

PII

개인 식별 정보를 참조하십시오.

플레이북

클라우드에서 핵심 운영 기능을 제공하는 등 마이그레이션과 관련된 작업을 캡처하는 일련의 사전 정의된 단계입니다. 플레이북은 스크립트, 자동화된 런북 또는 현대화된 환경을 운영하는 데 필요한 프로세스나 단계 요약의 형태를 취할 수 있습니다.

정책

권한을 정의 (ID 기반 정책 참조) 하거나, 액세스 조건을 지정 (리소스 기반 정책 참조) 하거나, 조직의 모든 계정에 대한 최대 권한을 정의 AWS Organizations (서비스 제어 정책 참조) 할 수 있는 개체입니다.

다국어 지속성

데이터 액세스 패턴 및 기타 요구 사항을 기반으로 독립적으로 마이크로서비스의 데이터 스토리지 기술 선택. 마이크로서비스가 동일한 데이터 스토리지 기술을 사용하는 경우 구현 문제가 발생하거나 성능이 저하될 수 있습니다. 요구 사항에 가장 적합한 데이터 스토어를 사용하면 마이크로서비스를 더 쉽게 구현하고 성능과 확장성을 높일 수 있습니다. 자세한 내용은 마이크로서비스에서 데이터 지속성 활성화를 참조하세요.

포트폴리오 평가

마이그레이션을 계획하기 위해 애플리케이션 포트폴리오를 검색 및 분석하고 우선순위를 정하는 프로세스입니다. 자세한 내용은 마이그레이션 준비 상태 평가를 참조하세요.

조건자

일반적으로 조항에 있는 `true` `false` OR를 반환하는 쿼리 조건입니다. `WHERE`

조건부 푸시다운

전송하기 전에 쿼리의 데이터를 필터링하는 데이터베이스 쿼리 최적화 기법입니다. 이렇게 하면 관계형 데이터베이스에서 검색하고 처리해야 하는 데이터의 양이 줄어들고 쿼리 성능이 향상됩니다.

예방적 제어

이벤트 발생을 방지하도록 설계된 보안 제어입니다. 이 제어는 네트워크에 대한 무단 액세스나 원치 않는 변경을 방지하는 데 도움이 되는 1차 방어선입니다. 자세한 내용은 Implementing security controls on AWS의 Preventative controls를 참조하세요.

보안 주체

작업을 수행하고 리소스에 액세스할 수 있는 AWS의 객체입니다. 이 엔터티는 일반적으로 AWS 계정의 루트 사용자, IAM 역할 또는 사용자입니다. 자세한 내용은 IAM 설명서의 역할 용어 및 개념의 보안 주체를 참조하세요.

개인 정보 보호 중심 설계

전체 엔지니어링 프로세스에서 개인 정보를 고려하는 시스템 엔지니어링에서의 접근 방식입니다.

프라이빗 호스팅 영역

Amazon Route 53에서 하나 이상의 VPC 내 도메인과 하위 도메인에 대한 DNS 쿼리에 응답하는 방법에 대한 정보가 담긴 컨테이너입니다. 자세한 내용은 Route 53 설명서의 프라이빗 호스팅 영역 작업을 참조하세요.

사전 예방적 제어

규정을 준수하지 않는 리소스의 배포를 방지하도록 설계된 보안 제어입니다. 이러한 컨트롤은 리소스를 프로비저닝하기 전에 리소스를 스캔합니다. 리소스가 컨트롤과 호환되지 않으면 프로비저닝되지 않습니다. 자세한 내용은 AWS Control Tower 설명서의 컨트롤 참조 안내서를 참조하고 보안 제어 구현의 사전 제어를 참조하십시오. AWS

프로덕션 환경

환경을 참조하십시오.

가명화

데이터세트의 개인 식별자를 자리 표시자 값으로 바꾸는 프로세스입니다. 가명화는 개인 정보를 보호하는 데 도움이 될 수 있습니다. 가명화된 데이터는 여전히 개인 데이터로 간주됩니다.

Q

쿼리 계획

SQL 관계형 데이터베이스 시스템의 데이터에 액세스하는 데 사용되는 일련의 단계 (예: 지침).

쿼리 계획 회귀

데이터베이스 서비스 최적화 프로그램이 데이터베이스 환경을 변경하기 전보다 덜 최적의 계획을 선택하는 경우입니다. 통계, 제한 사항, 환경 설정, 쿼리 파라미터 바인딩 및 데이터베이스 엔진 업데이트의 변경으로 인해 발생할 수 있습니다.

R

RACI 매트릭스

RACI (책임, 책임, 상담, 정보 제공) 를 참조하십시오.

랜섬웨어

결제가 완료될 때까지 컴퓨터 시스템이나 데이터에 대한 액세스를 차단하도록 설계된 악성 소프트웨어입니다.

RASCI 매트릭스

책임, 책임, 상담, 정보 제공 (RACI) 을 참조하십시오.

RCAC

행 및 열 액세스 제어를 참조하십시오.

읽기 전용 복제본

읽기 전용 용도로 사용되는 데이터베이스의 사본입니다. 쿼리를 읽기 전용 복제본으로 라우팅하여 기본 데이터베이스의 로드를 줄일 수 있습니다.

재설계

7 R을 참조하십시오.

Recovery Point Objective(RPO)

마지막 데이터 복구 시점 이후 허용되는 시간입니다. 이에 따라 마지막 복구 시점과 서비스 중단 시점 사이에 허용 가능한 데이터 손실이 결정됩니다.

Recovery Time Objective(RTO)

서비스 중단과 서비스 복구 사이에 허용되는 최대 지연 시간입니다.

리팩터링

7 R을 참조하십시오.

리전

AWS 리소스를 지리적 영역에 모아 놓은 것입니다. 각 AWS 리전은 내결함성, 안정성 및 복원력을 제공하기 위해 격리되어 있으며 다른 리전과는 독립적입니다. 자세한 내용은 AWS 일반 참조의 AWS 리전 관리를 참조하세요.

회귀

숫자 값을 예측하는 ML 기법입니다. 예를 들어, '이 집은 얼마에 팔릴까?'라는 문제를 풀기 위해 ML 모델은 선형 회귀 모델을 사용하여 주택에 대해 알려진 사실(예: 면적)을 기반으로 주택의 매매 가격을 예측할 수 있습니다.

리호스팅

7 R을 참조하십시오.

release

배포 프로세스에서 변경 사항을 프로덕션 환경으로 승격시키는 행위입니다.

고쳐 놓다

7 R을 참조하십시오.

리플랫포밍

7 R을 참조하십시오.

환매

7 R을 참조하십시오.

리소스 기반 정책

Amazon S3 버킷, 엔드포인트, 암호화 키 등의 리소스에 연결된 정책입니다. 이 유형의 정책은 액세스가 허용된 보안 주체, 지원되는 작업 및 충족해야 하는 기타 조건을 지정합니다.

RACI(Responsible, Accountable, Consulted, Informed) 매트릭스

마이그레이션 활동 및 클라우드 운영에 참여하는 모든 당사자의 역할과 책임을 정의하는 매트릭스입니다. 매트릭스 이름은 매트릭스에 정의된 책임 유형에서 파생됩니다: 실무 담당자 (R), 의사 결정권자 (A), 업무 수행 조언자 (C), 결과 통보 대상자 (I). 지원자는 (S) 선택사항입니다. 지원자를 포함하면 매트릭스를 RASCI 매트릭스라고 하고, 지원자를 제외하면 RACI 매트릭스라고 합니다.

대응 제어

보안 기준에서 벗어나거나 부정적인 이벤트를 해결하도록 설계된 보안 제어입니다. 자세한 내용은 Implementing security controls on AWS의 Responsive controls를 참조하세요.

retain

7 R을 참조하십시오.

은퇴

7 R을 참조하십시오.

회전

공격자가 자격 증명에 액세스하는 것을 더 어렵게 만들기 위해 암호를 주기적으로 업데이트하는 프로세스입니다.

행 및 열 액세스 제어(RCAC)

액세스 규칙이 정의된 기본적이고 유연한 SQL 표현식을 사용합니다. RCAC는 행 권한과 열 마스크로 구성됩니다.

RPO

복구 지점 목표를 참조하십시오.

RPO

복구 시간 목표를 참조하십시오.

런북

특정 작업을 수행하는 데 필요한 일련의 수동 또는 자동 절차입니다. 일반적으로 오류율이 높은 반복 작업이나 절차를 간소화하기 위해 런북을 만듭니다.

S

SAML 2.0

많은 ID 제공업체 (IdPs) 가 사용하는 개방형 표준입니다. 이 기능은 페더레이션형 AWS Single Sign-On(SSO)을 활성화하므로 조직의 모든 멤버에 대해 IAM 사용자를 생성하지 않아도 사용자가 AWS Management Console에 로그인하거나 AWS API 작업을 직접적으로 호출할 수 있습니다. SAML 2.0 기반 페더레이션에 대한 자세한 내용은 IAM 설명서의 SAML 2.0 기반 페더레이션 정보를 참조하세요.

SCP

서비스 제어 정책을 참조하십시오.

secret

에는 AWS Secrets Manager 암호화된 형태로 저장하는 비밀번호나 사용자 자격 증명과 같은 기밀 또는 제한된 정보. 비밀 값과 해당 메타데이터로 구성됩니다. 비밀 값은 바이너리, 단일 문자열 또는 여러 문자열일 수 있습니다. 자세한 내용은 Secrets Manager 문서의 시크릿을 참조하십시오.

보안 제어

위협 행위자가 보안 취약성을 악용하는 능력을 방지, 탐지 또는 감소시키는 기술적 또는 관리적 가드레일입니다. 보안 제어에는 예방적, 탐정적, 대응적, 사전 예방적 등 네 가지 기본 유형이 있습니다.

보안 강화

공격 표면을 줄여 공격에 대한 저항력을 높이는 프로세스입니다. 더 이상 필요하지 않은 리소스 제거, 최소 권한 부여의 보안 모범 사례 구현, 구성 파일의 불필요한 기능 비활성화 등의 작업이 여기에 포함될 수 있습니다.

보안 정보 및 이벤트 관리(SIEM) 시스템

보안 정보 관리(SIM)와 보안 이벤트 관리(SEM) 시스템을 결합하는 도구 및 서비스입니다. SIEM 시스템은 서버, 네트워크, 디바이스 및 기타 소스에서 데이터를 수집, 모니터링 및 분석하여 위협과 보안 침해를 탐지하고 알림을 생성합니다.

보안 대응 자동화

보안 이벤트에 자동으로 대응하거나 보안 이벤트를 해결하도록 설계된 사전 정의되고 프로그래밍된 조치입니다. 이러한 자동화는 보안 모범 사례를 구현하는 데 도움이 되는 탐지 또는 대응형 보안 제어 역할을 합니다. AWS 자동 응답 조치의 예로는 VPC 보안 그룹 수정, Amazon EC2 인스턴스 패치, 자격 증명 교체 등이 있습니다.

서버측 암호화

데이터를 수신하는 AWS 서비스가 대상에서 데이터를 암호화하는 것입니다.

서비스 제어 정책(SCP)

AWS Organizations에 속한 조직의 모든 계정에 대한 권한을 중앙 집중식으로 제어하는 정책입니다. SCP는 관리자가 사용자 또는 역할에 위임할 수 있는 작업에 대해 제한을 설정하거나 가드레일을 정의합니다. SCP를 허용 목록 또는 거부 목록으로 사용하여 허용하거나 금지할 서비스 또는 작업을 지정할 수 있습니다. 자세한 내용은 AWS Organizations 설명서의 서비스 제어 정책을 참조하세요.

서비스 엔드포인트

AWS 서비스에 대한 진입점의 URL입니다. 엔드포인트를 사용하여 대상 서비스에 프로그래밍 방식으로 연결할 수 있습니다. 자세한 내용은 AWS 일반 참조의 AWS 서비스 엔드포인트를 참조하세요.

서비스 수준에 관한 계약(SLA)

IT 팀이 고객에게 제공하기로 약속한 내용(예: 서비스 가동 시간 및 성능)을 명시한 계약입니다.

서비스 수준 지표 (SLI)

오류율, 가용성 또는 처리량과 같은 서비스의 성능 측면을 측정하는 것입니다.

서비스 수준 목표 (SLO)

서비스 수준 지표로 측정되는 서비스 상태를 나타내는 대상 지표입니다.

공동 책임 모델

클라우드 보안 및 규정 준수를 위해 AWS와 공유하는 책임을 설명하는 모델입니다. AWS는 클라우드의 보안을 담당하고, 사용자는 클라우드에서 보안을 담당합니다. 자세한 내용은 공동 책임 모델을 참조하세요.

시엠

보안 정보 및 이벤트 관리 시스템을 참조하십시오.

단일 장애 지점 (SPOF)

응용 프로그램의 중요한 단일 구성 요소에서 발생한 오류로 인해 시스템이 중단될 수 있습니다.

SLA

SLA (서비스 수준 계약) 를 참조하십시오.

SLI

서비스 수준 표시기를 참조하십시오.

SLO

서비스 수준 목표를 참조하십시오.

split-and-seed 모델

현대화 프로젝트를 확장하고 가속화하기 위한 패턴입니다. 새로운 기능과 제품 릴리스가 정의되면 핵심 팀이 분할되어 새로운 제품 팀이 만들어집니다. 이를 통해 조직의 역량과 서비스 규모를 조정하고, 개발자 생산성을 개선하고, 신속한 혁신을 지원할 수 있습니다. 자세한 내용은 의 애플리케이션 현대화를 위한 단계별 접근 방식을 참조하십시오. AWS 클라우드

SPOF

단일 장애 지점 보기

스타 스키마

하나의 큰 팩트 테이블을 사용하여 트랜잭션 또는 측정 데이터를 저장하고 하나 이상의 작은 차원 테이블을 사용하여 데이터 속성을 저장하는 데이터베이스 구성 구조입니다. 이 구조는 데이터 웨어하우스에서 사용하거나 비즈니스 인텔리전스 용도로 설계되었습니다.

Strangler Fig 패턴

레거시 시스템을 폐기할 수 있을 때까지 시스템 기능을 점진적으로 다시 작성하고 교체하여 모놀리식 시스템을 현대화하기 위한 접근 방식. 이 패턴은 무화과 덩굴이 나무로 자라 결국 숙주를 압도하고 대체하는 것과 비슷합니다. Martin Fowler가 모놀리식 시스템을 다시 작성할 때 위험을 관리하는 방법으로 이 패턴을 도입했습니다. 이 패턴을 적용하는 방법의 예는 컨테이너 및 Amazon API Gateway를 사용하여 기존의 Microsoft ASP.NET(ASMX) 웹 서비스를 점진적으로 현대화하는 방법을 참조하세요.

서브넷

VPC의 IP 주소 범위입니다. 서브넷은 단일 가용 영역에 상주해야 합니다.

대칭 암호화

동일한 키를 사용하여 데이터를 암호화하고 복호화하는 암호화 알고리즘입니다.

합성 테스트

잠재적 문제를 감지하거나 성능을 모니터링하기 위해 사용자 상호 작용을 시뮬레이션하는 방식으로 시스템을 테스트합니다. Amazon CloudWatch Synthetics를 사용하여 이러한 테스트를 생성할 수 있습니다.

T

tags

AWS 리소스를 구성하는 메타데이터 역할을 하는 키-값 페어입니다. 태그를 사용하면 리소스를 손쉽게 관리, 식별, 정리, 검색 및 필터링할 수 있습니다. 자세한 내용은 AWS 리소스에 태그 지정을 참조하십시오.

대상 변수

지도 ML에서 예측하려는 값으로, 결과 변수라고도 합니다. 예를 들어, 제조 설정에서 대상 변수는 제품 결함일 수 있습니다.

작업 목록

런북을 통해 진행 상황을 추적하는 데 사용되는 도구입니다. 작업 목록에는 런북의 개요와 완료해야 할 일반 작업 목록이 포함되어 있습니다. 각 일반 작업에 대한 예상 소요 시간, 소유자 및 진행 상황이 작업 목록에 포함됩니다.

테스트 환경

환경을 참조하십시오.

훈련

ML 모델이 학습할 수 있는 데이터를 제공하는 것입니다. 훈련 데이터에는 정답이 포함되어야 합니다. 학습 알고리즘은 훈련 데이터에서 대상(예측하려는 답)에 입력 데이터 속성을 매핑하는 패턴을 찾고, 이러한 패턴을 캡처하는 ML 모델을 출력합니다. 그런 다음 ML 모델을 사용하여 대상을 모르는 새 데이터에 대한 예측을 할 수 있습니다.

전송 게이트웨이

VPC와 온프레미스 네트워크를 상호 연결하는 데 사용할 수 있는 네트워크 전송 허브입니다. 자세한 내용은 AWS Transit Gateway 설명서의 Transit Gateway란 무엇입니까? 섹션을 참조하세요.

트렁크 기반 워크플로

개발자가 기능 브랜치에서 로컬로 기능을 구축하고 테스트한 다음 해당 변경 사항을 기본 브랜치에 병합하는 접근 방식입니다. 이후 기본 브랜치는 개발, 프로덕션 이전 및 프로덕션 환경에 순차적으로 구축됩니다.

신뢰할 수 있는 액세스

사용자를 대신하여 AWS Organizations의 조직과 해당 계정에서 작업을 수행하도록 지정하는 서비스에 권한을 부여하는 것입니다. 신뢰할 수 있는 서비스는 필요할 때 각 계정에 서비스 연결 역할을 생성하여 관리 작업을 수행합니다. 자세한 내용은 AWS Organizations 설명서의 다른 AWS 서비스와 함께 AWS Organizations 사용을 참조하세요.

튜닝

ML 모델의 정확도를 높이기 위해 훈련 프로세스의 측면을 여러 변경하는 것입니다. 예를 들어, 레이블링 세트를 생성하고 레이블을 추가한 다음 다양한 설정에서 이러한 단계를 여러 번 반복하여 모델을 최적화하는 방식으로 ML 모델을 훈련할 수 있습니다.

피자 두 판 팀

피자 두 판이면 먹을 수 있는 소규모 DevOps 팀이죠. 피자 두 판 팀 규모는 소프트웨어 개발에 있어 가능한 최상의 공동 작업 기회를 보장합니다.

U

불확실성

예측 ML 모델의 신뢰성을 저해할 수 있는 부정확하거나 불완전하거나 알려지지 않은 정보를 나타내는 개념입니다. 불확실성에는 두 가지 유형이 있습니다. 인식론적 불확실성은 제한적이고 불완전한 데이터에 의해 발생하는 반면, 우연한 불확실성은 데이터에 내재된 노이즈와 무작위성에 의해 발생합니다. 자세한 내용은 Quantifying uncertainty in deep learning systems 가이드를 참조하세요.

차별화되지 않은 작업

애플리케이션을 만들고 운영하는 데 필요하지만 최종 사용자에게 직접적인 가치를 제공하거나 경쟁 우위를 제공하지 못하는 작업을 헤비 리프팅이라고도 합니다. 차별화되지 않은 작업의 예로는 조달, 유지보수, 용량 계획 등이 있습니다.

상위 환경

환경을 보세요.

V

정리

스토리지를 회수하고 성능을 향상시키기 위해 증분 업데이트 후 정리 작업을 수반하는 데이터베이스 유지 관리 작업입니다.

버전 제어

리포지토리의 소스 코드 변경과 같은 변경 사항을 추적하는 프로세스 및 도구입니다.

VPC 피어링

프라이빗 IP 주소를 사용하여 트래픽을 라우팅할 수 있게 하는 두 VPC 간의 연결입니다. 자세한 내용은 Amazon VPC 설명서의 VPC 피어링이란?을 참조하세요.

취약성

시스템 보안을 손상시키는 소프트웨어 또는 하드웨어 결함입니다.

W

웜 캐시

자주 액세스하는 최신 관련 데이터를 포함하는 버퍼 캐시입니다. 버퍼 캐시에서 데이터베이스 인스턴스를 읽을 수 있기 때문에 주 메모리나 디스크에서 읽는 것보다 빠릅니다.

웜 데이터

자주 액세스하지 않는 데이터입니다. 이런 종류의 데이터를 쿼리할 때는 일반적으로 적절히 느린 쿼리가 허용됩니다.

윈도우 함수

현재 레코드와 어떤 식으로든 관련된 행 그룹에 대해 계산을 수행하는 SQL 함수입니다. 윈도우 함수는 이동 평균을 계산하거나 현재 행의 상대적 위치를 기반으로 행 값에 액세스하는 등의 작업을 처리하는 데 유용합니다.

워크로드

고객 대면 애플리케이션이나 백엔드 프로세스 같이 비즈니스 가치를 창출하는 리소스 및 코드 모음입니다.

워크스트림

마이그레이션 프로젝트에서 특정 작업 세트를 담당하는 직무 그룹입니다. 각 워크스트림은 독립적이지만 프로젝트의 다른 워크스트림을 지원합니다. 예를 들어, 포트폴리오 워크스트림은 애플리케이션 우선순위 지정, 웨이브 계획, 마이그레이션 메타데이터 수집을 담당합니다. 포트폴리오 워크스트림은 이러한 자산을 마이그레이션 워크스트림에 전달하고, 마이그레이션 워크스트림은 서버와 애플리케이션을 마이그레이션합니다.

웜

한 번 쓰고, 많이 읽으세요.

WQF

AWS 워크로드 검증 프레임워크를 참조하십시오.

한 번 작성하고 여러 번 읽기 (WORM)

데이터를 한 번 쓰고 데이터가 삭제되거나 수정되지 않도록 하는 스토리지 모델입니다. 인증된 사용자는 필요한 만큼 데이터를 여러 번 읽을 수 있지만 변경할 수는 없습니다. 이 데이터 스토리지 인프라는 변경할 수 없는 것으로 간주됩니다.

Z

제로데이 익스플로잇

제로데이 취약점을 악용하는 공격 (일반적으로 멀웨어) 입니다.

제로데이 취약성

프로덕션 시스템의 명백한 결함 또는 취약성입니다. 위협 행위자는 이러한 유형의 취약성을 사용하여 시스템을 공격할 수 있습니다. 개발자는 공격의 결과로 취약성을 인지하는 경우가 많습니다.

좀비 애플리케이션

평균 CPU 및 메모리 사용량이 5% 미만인 애플리케이션입니다. 마이그레이션 프로젝트에서는 이러한 애플리케이션을 사용 중지하는 것이 일반적입니다.

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