Praktik terbaik untuk menerapkan SQL Server di Amazon EC2

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Praktik terbaik untuk menerapkan SQL Server di Amazon EC2

Table of Contents

Praktik terbaik untuk menerapkan Microsoft SQL Server di Amazon EC2

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

Desember 2023 (riwayat dokumen)

Tujuan dari panduan ini adalah untuk memastikan pengalaman yang konsisten setelah menerapkan atau memigrasikan Microsoft SQL Server ke Amazon Elastic Compute Cloud (Amazon EC2) di Amazon Web Services () Cloud. AWS Ini memberikan praktik terbaik untuk mengonfigurasi database dan server Anda, untuk membantu mengoptimalkan infrastruktur Anda, menyesuaikan kinerja, dan menghindari masalah yang tidak terduga setelah penerapan atau migrasi.

Panduan ini ditujukan untuk arsitek database, sistem, dan prospek database, serta administrator yang berencana untuk memigrasi Microsoft SQL Server dari lingkungan lokal mereka ke Amazon EC2, atau yang ingin mengoptimalkan penerapan SQL Server baru mereka di Amazon EC2.

Amazon EC2 menyediakan kapasitas komputasi yang dapat diskalakan di Cloud. AWS Menggunakan SQL Server di Amazon EC2 mirip dengan menjalankan SQL Server di tempat. Amazon EC2 memberi Anda kontrol penuh atas infrastruktur dan lingkungan database Anda. Anda mendapat manfaat dari skala, kinerja, dan elastisitas AWS Cloud, tetapi Anda bertanggung jawab untuk mengonfigurasi dan menyetel semua komponen, termasuk instans EC2, volume penyimpanan, sistem file, jaringan, dan keamanan. Panduan ini memberikan informasi untuk membantu Anda mengoptimalkan konfigurasi dan memaksimalkan kinerja SQL Server. AWS Ini membahas pengaturan server dan penyimpanan dan praktik terbaik secara rinci. Ini juga menjelaskan cara mengotomatiskan pengaturan jika berlaku, dan membahas perubahan konfigurasi di tingkat database.

> **Note**
>
> AWSjuga menawarkan opsi untuk memindahkan database SQL Server lokal Anda ke layanan terkelola seperti Amazon Relational Database Service (Amazon RDS) untuk SQL Server. Untuk diskusi tentang opsi migrasi, lihat Strategi migrasi untuk database relasional di situs web Panduan AWS Preskriptif.

Mengkonfigurasi pengaturan komputasi dan penyimpanan

Sebelum Anda memigrasi atau menerapkan SQL Server di Amazon EC2, Anda dapat mengonfigurasi instans EC2 dan pengaturan penyimpanan untuk meningkatkan kinerja dan menurunkan biaya. Bagian berikut memberikan kiat pengoptimalan dan praktik terbaik.

Topik

Menggunakan jenis instans yang dioptimalkan Amazon EBS

Jika database SQL Server Anda menangani beban kerja I/O-intensif, penyediaanInstans yang dioptimalkan Amazon Elastic Block Store (Amazon EBS)akan membantu meningkatkan kinerja.

Instans yang dioptimalkan Amazon EBS menggunakan tumpukan konfigurasi yang dioptimalkan dan memberikan kapasitas khusus tambahan untuk I/O Amazon EBS.

Optimalkan tata letak disk atau distribusi file

Gunakan satu volume untuk data dan file log, volume lain untuk beban kerja tempdb, dan Cold HDD (`sc1`) atau Throughput HDD yang Dioptimalkan (`st1`) volume untuk backup.

Jika Anda mengalami masalah terkait I/O dan ingin memisahkan beban kerja untuk data dan file log, pertimbangkan untuk menggunakan volume yang berbeda. Jika beban kerja Anda mengharuskan Anda untuk memisahkan database tertentu, pertimbangkan untuk menggunakan volume khusus untuk setiap database.

Biasanya, tempdb adalah target I/O tertinggi, jadi jika beban kerja itu tidak dipisahkan, itu mungkin menjadi hambatan. Pemisahan ini juga membantu mengisolasi tempdb dari data dan log file database pengguna. Anda dapat menggunakan penyimpanan dengan biaya yang relatif lebih rendah untuk pencadangan guna mengoptimalkan biaya Anda.

Mengatur ukuran unit alokasi NTFS 64 KB

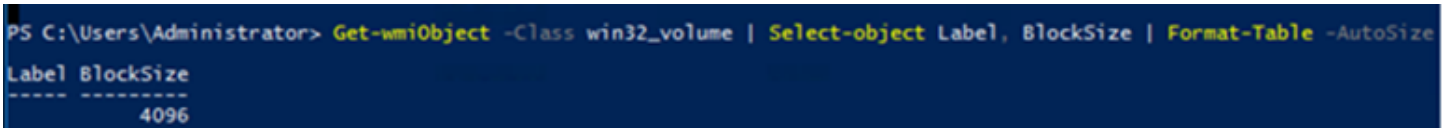
Unit atom penyimpanan di SQL Server adalahhalaman, yang berukuran 8 KB. Delapan halaman yang bersebelahan secara fisik membentukseluas(yang berukuran 64 KB). SQL Server menggunakan luasan untuk menyimpan data. Oleh karena itu, pada mesin SQL Server, ukuran unit alokasi NTFS untuk hosting file database SQL (termasuk tempdb) harus 64 KB.

Untuk memeriksa ukuran cluster (alokasi NTFS) drive Anda, Anda dapat menggunakanPowerShellatau baris perintah.

MenggunakanPowerShell:

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

Ilustrasi berikut menunjukkan contoh output dariPowerShell.



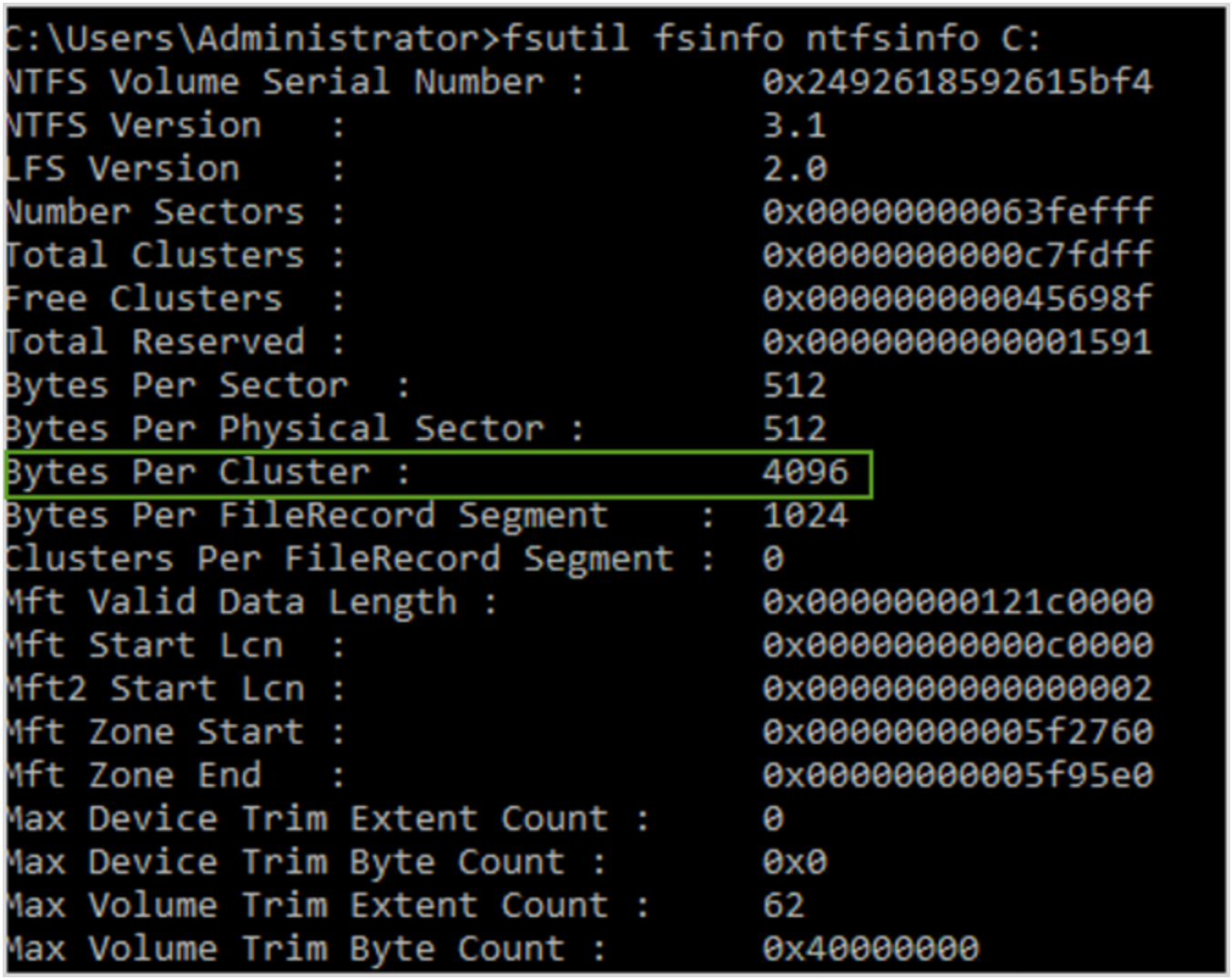
Atau gunakan:

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

Menggunakan baris perintah:

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

Ilustrasi berikut menunjukkan contoh output dari baris perintah. YangByte Per Clusternilai menampilkan ukuran format dalam byte. Contoh output menunjukkan 4096 byte. Untuk drive yang meng-host berkas database SQL Server, nilai ini harus 64 KB.



Dalam beberapa kasus, kinerja SQL Server tidak bergantung pada ukuran blok saat Anda menggunakan penyimpanan SSD di Amazon EC2. Untuk informasi lebih lanjut, lihat posting blogLakukanAWSpelanggan mendapat manfaat dari ukuran blok 64KB untuk penyimpanan SQL Server?

Tempatkan tempdb di toko instance

Saat Anda menggunakan penyimpanan instans Amazon EC2, gunakan volume penyimpanan instans untuk tempdb. Toko instans menyediakan penyimpanan tingkat blok sementara (sementara) untuk instans Anda. Kami menyarankan Anda menempatkan tempdb pada volume penyimpanan

instans karena dua alasan: kecepatan dan biaya. TemPDb biasanya database yang paling banyak digunakan, sehingga manfaat dari drive tercepat yang tersedia. Manfaat lain dari menempatkan tempdb di toko instans adalah penghematan biaya, karena Anda tidak dikenakan biaya secara terpisah untuk I/O terhadap toko instans.

TemPDb diciptakan setiap kali Anda me-restart SQL Server, sehingga menghentikan atau mengakhiri instance tidak akan menyebabkan kehilangan data. Namun, volume toko instance hilang ketika mesin virtual dimulai pada host lain, karena disk sementara terpasang secara lokal ke mesin, jadi rencanakan dengan hati-hati.

Saat Anda menggunakan volume penyimpanan instans:

- Inisialisasi volume sebelum layanan SQL Server dimulai. Jika tidak, prosedur startup SQL Server akan gagal.
- Berikan izin (kontrol penuh) pada volume penyimpanan instans secara eksplisit ke akun startup SQL Server.

Memindahkan tempdb ke toko instance

Untuk memindahkan tempdb ke volume penyimpanan instance:

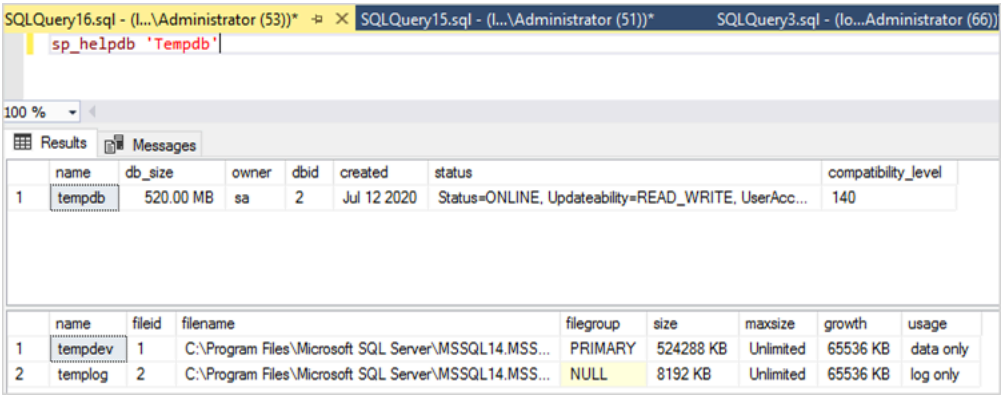
1. Dari Windows, jalankan`diskmgmt.msc`sebagai administrator untuk membuka utilitas sistem Manajemen Disk.
2. Inisialisasi disk baru.
3. Klik kanan disk, lalu pilihVolume Sederhana Baru.
4. Selesaikan petunjuknya, gunakan pengaturan ini untuk memformat volume:

   - Sistem file: NTFS
   - Ukuran unit alokasi: 64K
   - Label volume: tempdb

   Untuk informasi lebih lanjut, lihatDokumentasi Manajemen Diskdi situs web Microsoft.
5. Hubungkan ke contoh SQL Server, dan jalankan perintah berikut untuk mencatat nama file logis dan fisik dari database tempdb:

```
$ sp_helpdb 'tempdb'
```

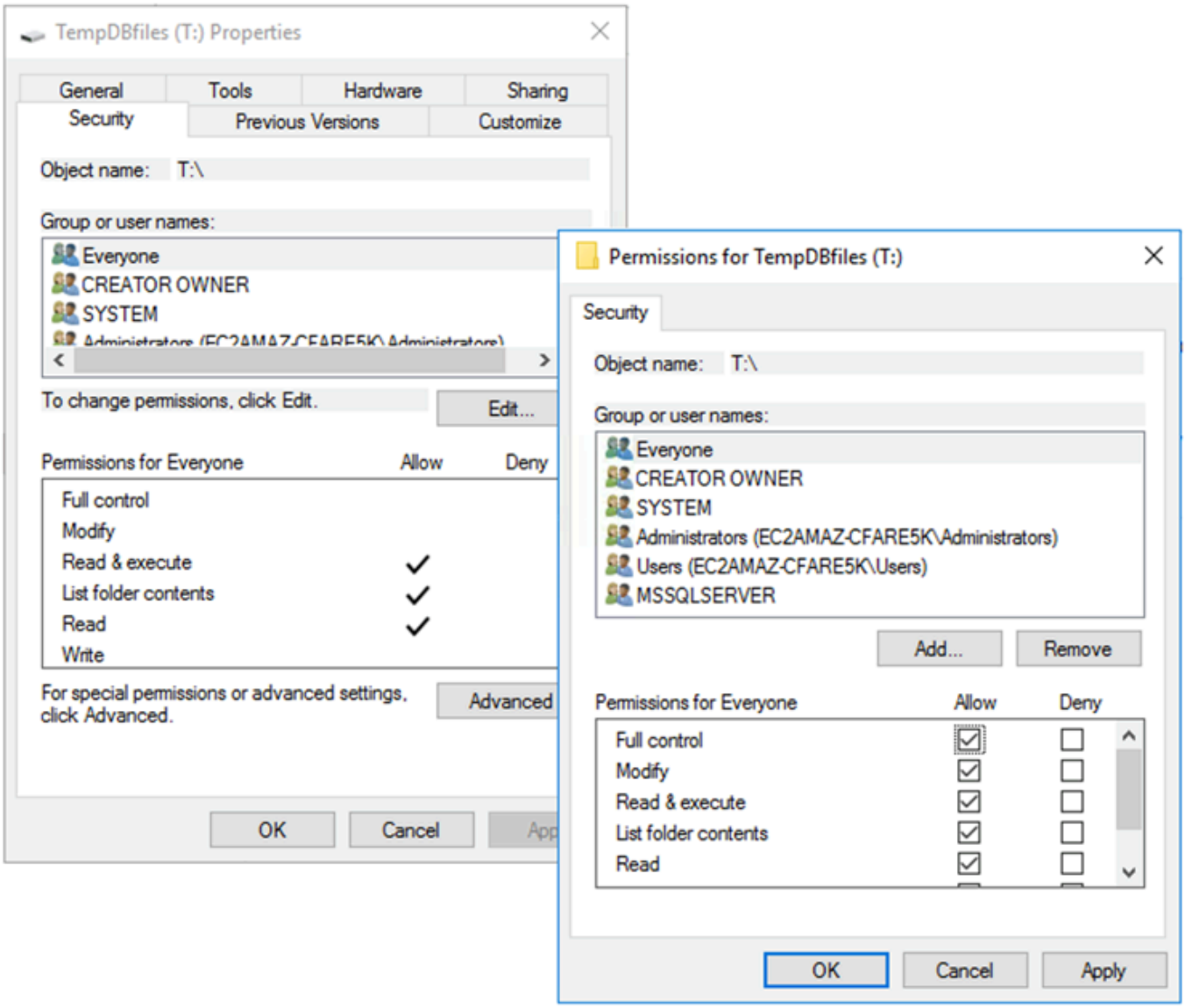
Screenshot berikut menunjukkan perintah dan outputnya.



6. Pindahkan file tempdb ke lokasi baru. Ingatlah untuk mengatur semua file database tempdb ke ukuran awal yang sama. Contoh script SQL server berikut bergerak file tempdb untuk mendorong T dan menetapkan file data ke ukuran yang sama.

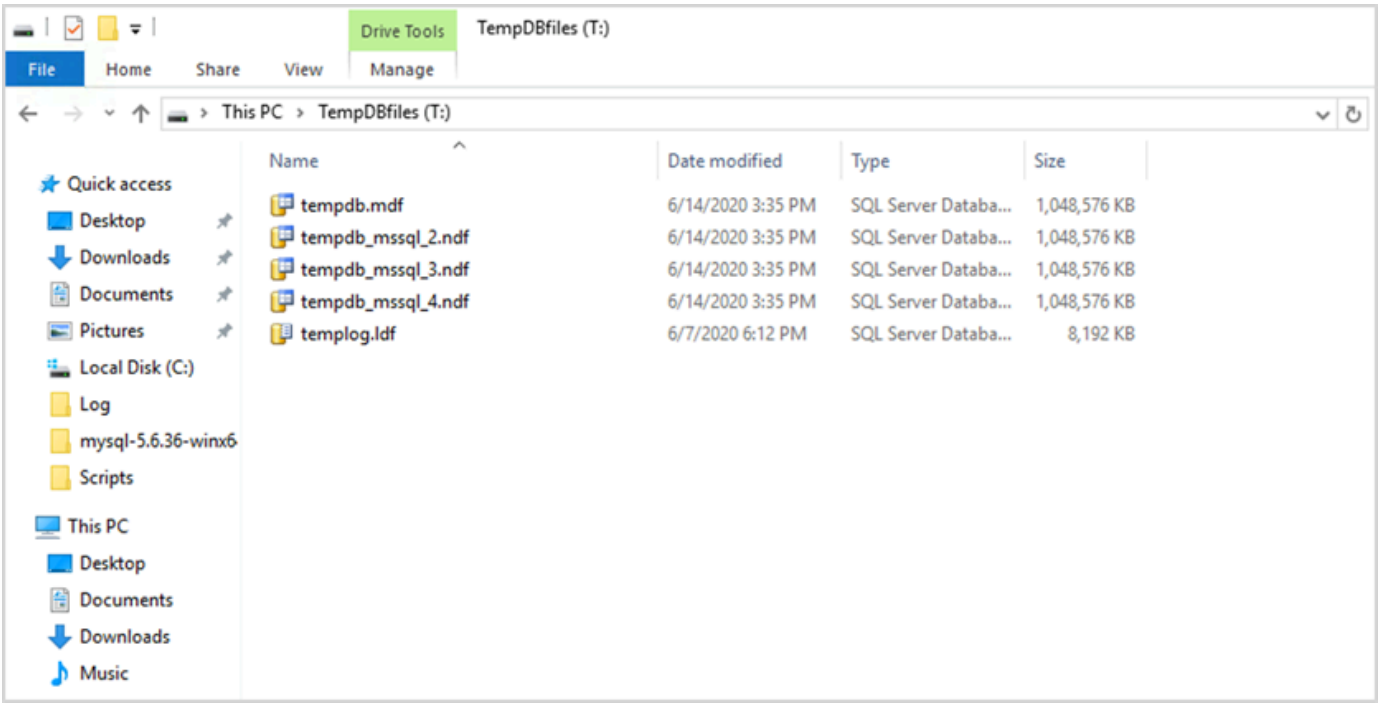
```
USE master
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = tempdev, FILENAME = 'T:\tempdb.mdf',SIZE
 = 524288KB)
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = temp2, FILENAME = 'T:
\tempdb_mssql_2.ndf',SIZE = 524288KB)
GO
ALTER DATABASE TempDB MODIFY FILE (NAME = temp3, FILENAME = 'T:
\tempdb_mssql_3.ndf',SIZE = 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. Berikan izin akun startup SQL Server ke lokasi baru database tempdb, sehingga dapat membuat file tempdb, seperti yang ditunjukkan pada gambar berikut.



8. Mulai ulang SQL Server untuk menggunakan lokasi baru untuk tempdb.

   Anda akan melihat file tempdb dibuat di lokasi baru, seperti yang ditunjukkan pada gambar berikut.



9. Hapus file tempdb dari lokasi lama.

Untuk memastikan bahwa volume penyimpanan instance diinisialisasi sebelum SQL Server dimulai dalam kasus reboot atau start/stop, ikuti langkah-langkah di bagian berikutnya. Jika tidak, startup SQL Server akan gagal karena tempdb tidak diinisialisasi.

Menginisialisasi toko instance

Untuk menginisialisasi penyimpanan data:

1. Buka Manajer Layanan Windows (`services.msc`), dan mengatur SQL Server dan layanan dependennya (misalnya, SQL Server Agent) untuk memulai secara manual. (Anda akan menggunakan skrip untuk memulainya ketika volume penyimpanan instance sudah siap.)
2. Buat sebuahPowerShellskrip untuk diteruskan ke instans Amazon EC2 sebagai data pengguna. Script ini melakukan hal berikut:
   - Mendeteksi penyimpanan sementara dan menciptakan tempdb drive untuk itu (drive T dalam contoh).
   - Menyegarkan disk fana jika instans EC2 berhenti dan restart.
   - Memberikan akun startup SQL Server kontrol penuh dari volume tempdb yang baru diinisialisasi. Contoh mengasumsikan contoh default, sehingga menggunakan`NT SERVICE`

\MSSQLSERVER. Untuk contoh bernama, ini biasanya akanNT SERVICE\MSSQL$*<InstanceName>*secara default.

- Menyimpan skrip pada volume lokal (`c:\scripts`dalam contoh) dan memberikan nama file (`InstanceStoreMapping.ps1`).
- Membuat tugas yang dijadwalkan menggunakan Windows Task Scheduler. Tugas ini menjalankanPowerShellscript pada startup.
- Mulai SQL Server dan SQL Server Agent setelah tindakan sebelumnya.

Script berikut adalah dari lab keduaLokakarya Grup Ketersediaan MS-SQLdengan beberapa perubahan. Salin skrip kePenggunabidang data saat Anda meluncurkan instans EC2, dan menyesuaikannya seperlunya.

```
<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>
```

Menggunakan ekstensi buffer pool

Jika Anda berencana untuk menggunakan ekstensi buffer pool, Anda mungkin juga mempertimbangkan menempatkannya pada volume sementara. Namun, kami sangat menyarankan untuk mengujinya secara menyeluruh sebelum implementasi. Hindari menggunakan volume yang sama untuk ekstensi buffer pool dan tempdb.

> **Note**
>
> Meskipun ekstensi buffer pool dapat berguna dalam beberapa kasus, itu bukan pengganti RAM. Sebelum Anda memutuskan untuk menggunakannya, lihatdetail yang disediakan di situs web Microsoft.

Hindari ketidakcocokan inti CPU

Memilih server yang memiliki jumlah core lebih tinggi daripada penutup lisensi Anda dapat menyebabkan CPU miring dan membuang daya CPU. Hal ini karena pemetaan antara core logis

dan aktual. Bila Anda menggunakan SQL Server dengan Client Access License (CAL), beberapa penjadwal akan`VISIBLE ONLINE`dan sisanya akan`VISIBLE OFFLINE`. Hal ini dapat menyebabkan masalah kinerja dengan topologi akses memori non-seragam (NUMA), karena node scheduler tidak digunakan secara optimal.

Misalnya, jika Anda menjalankan SQL Server pada`m5.24xlarge`misalnya, itu akan mendeteksi dua soket dengan 24 core, dan 48 prosesor logis per soket, yang menghasilkan total 96 prosesor logis. Jika Anda memiliki lisensi untuk hanya 48 core, Anda akan melihat pesan yang mirip dengan berikut ini di log galat SQL Server:

2020-06-08 12:35:27.37 Server SQL Server mendeteksi 2 soket dengan 24 core per soket dan 48 prosesor logis per soket, 96 total prosesor logis; menggunakan 48 prosesor logis berdasarkan lisensi SQL Server. Ini adalah pesan informasi; tidak diperlukan tindakan pengguna.

Jika Anda melihat perbedaan antara total core dan jumlah core yang digunakan oleh SQL Server, periksa ketidakseimbangan penggunaan CPU atau gunakan jenis server yang memiliki jumlah core yang sama dengan yang didukung lisensi Anda.

CPU miring:Untuk jenis instance dalam contoh kita (`m5.24xlarge`), SQL Server menciptakan delapan node NUMA secara default. Hanya empat node ini (parent node ID 0,1,2,3) yang memiliki penjadwal dengan status`VISIBLE ONLINE`. Jadwal yang tersisa adalah semua`VISIBLE OFFLINE`. Perbedaan antara penjadwal ini dapat menyebabkan penurunan kinerja.

Untuk memeriksa informasi dan status penjadwal, gunakan:

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

Jika Anda ingin menggunakan instance server yang memiliki jumlah core yang lebih tinggi daripada dukungan lisensi SQL Server Anda, pertimbangkan untuk menyesuaikan jumlah core dengan mengikuti petunjuk di[Menentukan opsi CPU untuk instans Anda](#)dalam dokumentasi Amazon EC2.

Uji kinerja disk

Kami menyarankan Anda memeriksa kinerja disk Anda dengan menggunakan alat seperti[DiskSpd](#). Alat ini memberi Anda perkiraan kecepatan disk sebelum Anda menjalankan pengujian khusus SQL Server. Sangat penting untuk mengevaluasi kinerja disk, karena volume EBS bekerja secara berbeda dari SAN tradisional di lingkungan lokal. Kurangnya pengujian kinerja yang tepat dapat menyebabkan kinerja lambat yang tidak terduga setelah migrasi. Anda juga dapat menjalankan[tes kustom](#)bersamaDiskSpd.

Aktifkan inisialisasi file instan

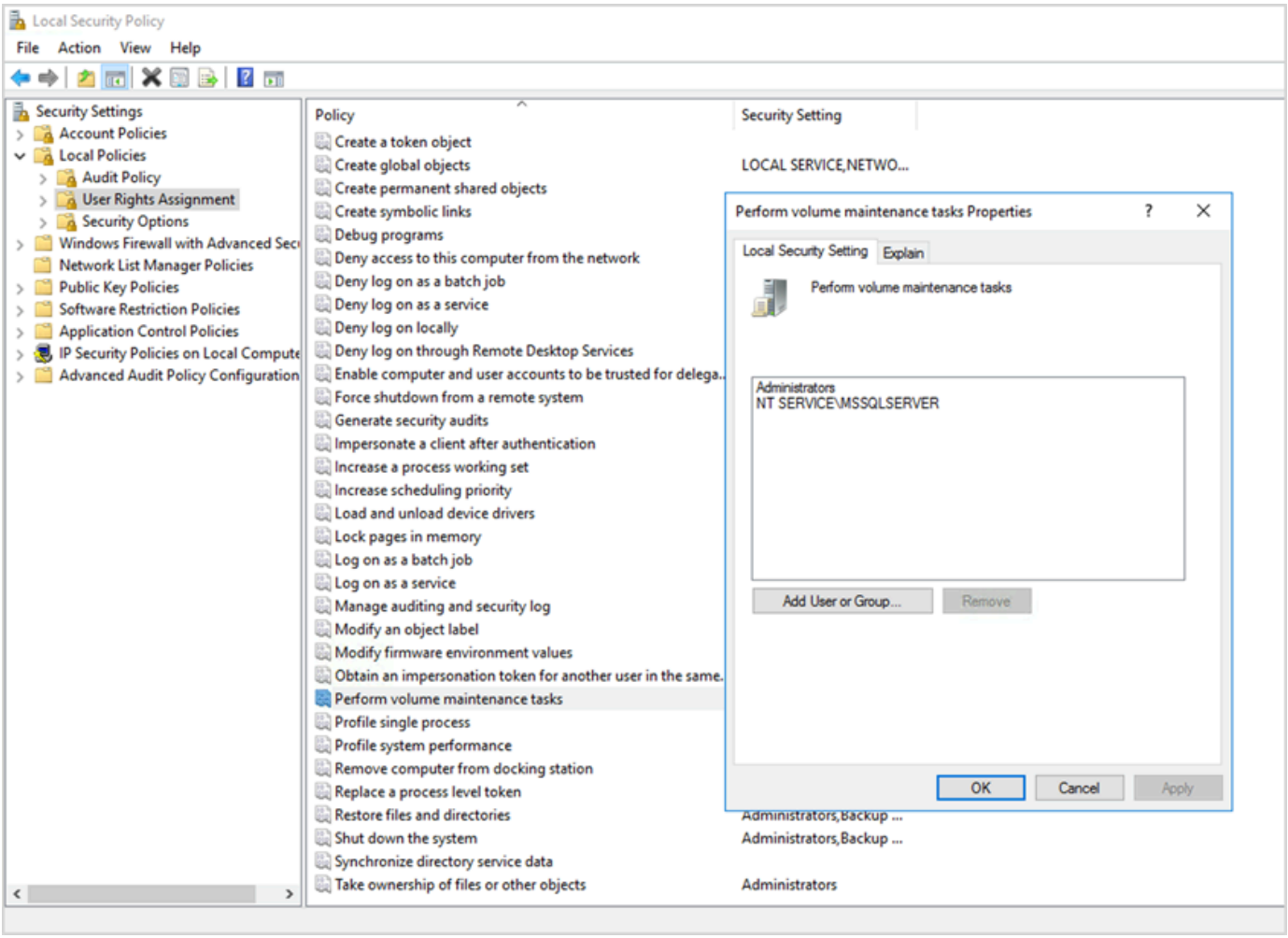
Di SQL Server, gunakanMelakukan tugas pemeliharaan volumepengaturan untuk mengaktifkan inisialisasi file instan, kecuali jika Anda mengikuti batasan peraturan. Opsi ini secara signifikan meningkatkan kinerja pertumbuhan otomatis file.

Pengaturan ini melewatkan operasi zeroing untuk file data. Artinya, file data tidak diisi dengan nilai nol (`0x0`) ketika mereka diinisialisasi, yang dapat memakan waktu lama. Konten saat ini pada disk ditimpa hanya ketika data baru ditulis ke disk.

> **Note**
>
> File log tidak mendapat manfaat dari inisialisasi file instan.

Untuk mengaktifkan inisialisasi file instan:

1. PadaMulailayar, jalankan`secpol.msc`untuk membukaKebijakan Keamanan Lokalkonsol.
2. PilihKebijakan Lokal,Penugasan Hak Pengguna,Melakukan tugas pemeliharaan volume, dan tambahkan akun layanan SQL Server, seperti yang ditunjukkan pada gambar berikut.



3. Mulai ulang contoh SQL Server agar perubahan diterapkan.

Untuk informasi selengkapnya tentang inisialisasi file instan, lihatDokumentasi SQL Serverdi situs web Microsoft.

> **Catatan keamanan**
>
> Bila Anda menggunakan inisialisasi file instan, disk ditimpa hanya ketika data baru ditulis ke file, jadi dimungkinkan untuk membaca konten yang dihapus.
> Sementara drive dilampirkan ke instance, daftar kontrol akses diskresioner (DACL) pada file mengurangi risiko pengungkapan informasi, karena memungkinkan akses hanya ke akun layanan SQL Server dan administrator lokal. Namun, ketika file terlepas, itu dapat diakses. Jika pengungkapan konten yang dihapus menjadi masalah, Anda harus menonaktifkan inisialisasi file instan untuk contoh SQL Server.

Kunci halaman dalam memori

AktifkanKunci halaman dalam memoripilihan untuk akun startup SQL Server untuk memastikan bahwa sistem operasi tidak memangkas set kerja SQL Server.

Untuk memeriksa apakah opsi ini diaktifkan, gunakan kueri SQL berikut:

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

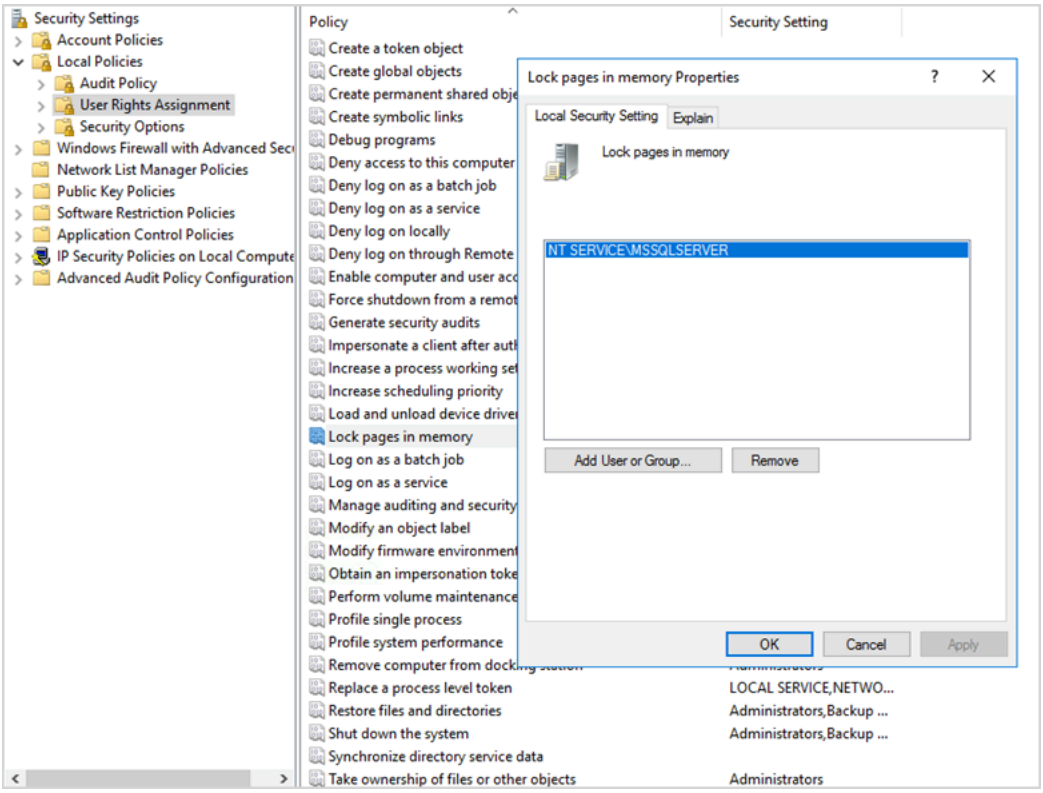
Output:

```
sql_memory_model     sql_memory_model_desc
1                   CONVENTIONAL

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

Untuk mengaktifkanKunci halaman dalam memoripilihan:

1. PadaMulailayar, jalankan`secpol.msc`untuk membukaKebijakan Keamanan Lokalkonsol.
2. PilihKebijakan Lokal,Penugasan Hak Pengguna,Kunci halaman dalam memori, dan tambahkan akun layanan SQL Server, seperti yang ditunjukkan pada gambar berikut.



3. Mulai ulang contoh SQL Server agar perubahan diterapkan.
4. Gunakan query SQL berikut untuk mengkonfirmasi bahwaKunci halaman dalam memoriopsi diaktifkan:

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

Output:

```
sql_memory_model     sql_memory_model_desc
2                    LOCK_PAGES

"LOCK_PAGES" means it’s enabled.
```

Untuk informasi selengkapnya tentang model memori SQL Server, lihat`sql_memory_model`dan`sql_memory_model_desc`di dalamdokumentasi sys.dm_os_sys_infodi situs web Microsoft.

Nonaktifkan pembongkaran TCP dan pengaturan RSS

Jika Anda mengamati masalah konektivitas acak seperti kesalahan tingkat transportasi atau kesalahan transmisi paket saat menjalankan beban kerja SQL, Anda mungkin ingin menonaktifkan pembongkaran TCP dan pengaturan RSS.

- Pembongkaran TCP (Offload Cerobong TCPfitur) membongkar pemrosesan paket TCP/IP dari prosesor ke adaptor jaringan, untuk membebaskan CPU untuk tugas-tugas lain.
- Receive-side scaling (RSS) membantu mendistribusikan pemrosesan lalu lintas jaringan masuk pada sistem multiprosesor. Ini memuat menyeimbangkan pemrosesan jaringan secara efisien di antara CPU.

Untuk memeriksa pengaturan Anda saat ini, pada prompt perintah, jalankannetshperintah:

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

Berikut adalah contoh output dari perintah. Dalam contoh ini,Status Penskalaan Sisi MenerimadanNegara Offload Cerobongkeduanya dinonaktifkan.

```
C:\Users\Administrator>netsh int tcp show global
Querying active state...

TCP Global Parameters
----------------------------------------------
Receive-Side Scaling State          : disabled
Chimney Offload State               : disabled
NetDMA State                        : disabled
Direct Cache Access (DCA)           : disabled
Receive Window Auto-Tuning Level    : normal
Add-On Congestion Control Provider  : none
ECN Capability                      : enabled
RFC 1323 Timestamps                 : disabled
Initial RTO                         : 3000
Receive Segment Coalescing State    : enabled
Non Sack Rtt Resiliency             : disabled
Max SYN Retransmissions             : 2
TCP Fast Open                       : disabled
```

Untuk mendapatkan informasi tugas offload tentang koneksi tertentu, pada prompt perintah, jalankan:

```
netstat -t
```

dan periksa nilai kolom status offload.

Untuk menonaktifkan pembongkaran TCP dan RSS untuk Windows Server 2008 dan 2012, jalankan perintah ini pada prompt perintah:

```
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
```

Untuk mempelajari lebih lanjut tentang pengaturan ini, lihat:

- Fitur TCP Chimney Offload, Menerima Penskalaan Samping, dan Akses Memori Langsung JaringandanPengantar Menerima Penskalaan Sisidi situs web Microsoft
- Pembongkaran TCPdalam dokumentasi Amazon EC2
- Pemecahan Masalah Driver PVdalam dokumentasi Amazon EC2

> **Important**
>
> Jangan gunakanIPsec Tugas OffloadatauOffload Cerobong TCP. MenurutDokumentasi Microsoft, fitur pembongkaran ini tidak berlaku lagi di Windows Server 2016 dan mungkin tidak didukung di versi mendatang. Menggunakan fitur-fitur ini dapat mempengaruhi kinerja.

Tentukan persyaratan IOPS dan throughput Anda

Gunakan Windows Performance Monitor untuk mendapatkan informasi tentang IOPS dan throughput.

Untuk membuka Windows Performance Monitor, jalankanperfmonpada prompt perintah. IOPS dan data throughput disediakan oleh penghitung kinerja berikut:

- Bacaan disk/dtk+disk menulis/detik = IOPS
- Disk membaca byte/dtk+disk menulis byte/detik = throughput

Kami menyarankan Anda mendapatkan IOPS dan data throughput untuk waktu penggunaan puncak dan juga selama siklus beban kerja biasa, untuk mendapatkan perkiraan yang baik dari kebutuhan Anda. Pastikan bahwa jenis instans yang Anda pilih untuk SQL Server mendukung persyaratan I/O ini.

Penting untuk mendapatkan perkiraan ini dengan benar. Jika tidak, Anda mungkin menyediakan sumber daya secara berlebihan, yang mungkin mengakibatkan sumber daya yang kurang dimanfaatkan, atau kekurangan penyediaan sumber daya Anda, yang mungkin mengakibatkan masalah kinerja yang parah.

Gunakan striping untuk melewati batasan IOPS dan throughput

Ketika aplikasi SQL Server Anda memerlukan lebih dariIOPS dan throughput maksimumtersedia pada volume EBS, pertimbangkan untuk melucuti volume EBS Anda untuk mengatasi keterbatasan ini.

Volume striping (RAID)membantu Anda memenuhi persyaratan IOPS dan throughput Anda, dan dibatasi oleh IOPS dan bandwidth maksimum yang didukung oleh instans tertentu. Untuk informasi selengkapnya tentang opsi striping, lihatKonfigurasi RAIDdalam dokumentasi Amazon EC2. Anda juga dapat menggunakan Storage Spaces di server mandiri. Untuk informasi lebih lanjut, lihatDokumentasi Microsoft.

Kecualikan file SQL Server dari perangkat lunak antivirus

Ketika Anda mengkonfigurasi pengaturan perangkat lunak antivirus Anda, pastikan bahwa Anda mengecualikan file SQL Server dan direktori dari pemindaian virus. Untuk detail dan daftar file dan direktori untuk dikecualikan, lihatCara memilih perangkat lunak antivirus untuk berjalan di komputer yang menjalankan SQL Serverdi situs web Microsoft.

Jika Anda tidak mengecualikan berkas SQL Server ini, mereka bisa rusak atau dikarantina oleh perangkat lunak antivirus ketika SQL Server perlu menggunakannya. Tidak termasuk file-file ini juga dapat menyebabkan masalah kinerja.

Mengkonfigurasi SQL Server

Bagian ini memberikan praktik terbaik untuk mengonfigurasi database SQL Server Anda untuk menyempurnakan kinerja, menghindari jebakan umum, dan memenuhi persyaratan keamanan dan ketersediaan Anda. Anda dapat menerapkan perubahan ini sebelum atau setelah memigrasikan database ke Amazon EC2. Bagian berikut memberikan tips konfigurasi dan praktik terbaik.

Topik

Konfigurasikan tempdb untuk mengurangi pertengkaran

Kami menyarankan Anda mengonfigurasi tempdb dengan beberapa file data dengan ukuran yang sama dan dengan faktor pertumbuhan yang sama.

Pada server database yang sibuk yang banyak menggunakan tempdb, Anda mungkin melihat pemblokiran parah ketika server mengalami beban berat. Anda mungkin memperhatikan bahwa tugas sedang menunggu sumber daya tunggu yang mengarah ke halaman di tempdb. Halaman-halaman ini mungkin halaman Page Free Space (PFS) dan Shared Global Allocation Map (SGM) yang memiliki format 2: *x*: *x* (misalnya, 2:1:1 atau 2:1:2).

Untuk meningkatkan konkurensi tempdb, Anda dapat meningkatkan jumlah file data dalam tempdb untuk memaksimalkan bandwidth disk dan mengurangi pertengkaran dalam struktur alokasi. Berikut adalah beberapa pedoman:

- Jika jumlah prosesor logis sama dengan, atau kurang dari, 8: Gunakan jumlah file data dan prosesor logis yang sama.
- Jika jumlah prosesor logis lebih tinggi dari 8: Gunakan 8 file data.

Jika pertengkaran berlanjut, tingkatkan jumlah file data dalam kelipatan 4 sampai pertengkaran diperbaiki, hingga jumlah prosesor logis di server. Ini akan membantu menghindari pertengkaran SGAM di tempdb. Jika Anda menggunakan SQL Server 2014 atau versi sebelumnya, Anda juga perlu mengaktifkan flag jejak 1118. Bendera ini memaksa alokasi halaman pada luasan seragam alih-alih luasan campuran, yang meminimalkan pemindaian pada halaman SGAM dan mengurangi pertengkaran.

Dimulai dengan SQL Server 2016 (13.x), perilaku ini dikendalikan oleh `AUTOGROW_SINGLE_FILE` dan `AUTOGROW_ALL_FILES` opsi. `ALTER DATABASE` Sebagai contoh:

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

Untuk informasi selengkapnya tentang menyetel opsi ini, lihat dokumentasi Microsoft SQL Server.

Atur MAXDOP untuk performa terbaik

Tingkat maksimum paralelisme (MAXDOP) adalah opsi konfigurasi server untuk menjalankan SQL Server pada beberapa CPU. Ini mengontrol jumlah prosesor yang digunakan untuk menjalankan pernyataan tunggal dalam eksekusi rencana paralel. Nilai default adalah 0, yang memungkinkan SQL Server untuk menggunakan semua prosesor yang tersedia. Ini dapat memengaruhi kinerja, dan tidak optimal untuk sebagian besar kasus penggunaan.

Gunakan panduan berikut saat Anda mengonfigurasi nilai MAXDOP untuk SQL Server.

Node NUMA	Prosesor logis	Nilai MAXDOP
Tunggal	≤ 8	4, 2, atau jumlah core (untuk satu atau dua core)
Tunggal	> 8	8, 4, atau 2
Beberapa	≤ 16	8, 4, atau 2

Node NUMA	Prosesor logis	Nilai MAXDOP
Beberapa	> 16	16, 8, 4, atau 2

> **Note**
>
> Mengatur MAXDOP ke 2, 4, atau 8 umumnya memberikan hasil terbaik dalam sebagian besar kasus penggunaan. Kami menyarankan Anda menguji beban kerja Anda dan memantau jenis tunggu terkait paralelisme seperti. `CXPACKET`

Anda dapat menggunakan kueri berikut untuk mengumpulkan konfigurasi NUMA saat ini untuk SQL Server 2016 dan versi yang lebih baru:

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

di mana:

- `cpu_count`mengacu pada jumlah CPU logis dalam sistem.
- `hyperthread_ratio`adalah rasio jumlah core yang diekspos oleh satu prosesor fisik.
- `softnuma_configuration`adalah`0`,`1`, atau`2`:
  - `0 (OFF)`: default
  - `1 (automated)`: Soft-numa
  - `2 (manual)`: Soft-numa
- `softnuma_configuration_desc`adalah`OFF`,`ON`, atau`MANUAL`:
  - `OFF`menunjukkan bahwa fitur Soft-numa tidak aktif.
  - `ON`menunjukkan bahwa SQL Server secara otomatis memutuskan ukuran node NUMA.

  - `MANUAL`menunjukkan bahwa Soft-numa dikonfigurasi secara manual.
- `socket_count`adalah jumlah soket prosesor.
- `numa_node_count`adalah jumlah node NUMA yang tersedia dalam sistem.

Untuk memeriksa nilai MAXDOP saat ini, gunakan:

```
$ sp_configure 'max_degree_of_parallelism'
```

Untuk informasi selengkapnya tentang MAXDOP, lihat dokumentasi Microsoft SQL Server.

Ubah ambang biaya paralelisme

Ambang biaya paralelisme menentukan kueri mana yang merupakan kandidat untuk eksekusi paralel. Nilai default properti ini adalah 5, yang berarti bahwa pengoptimal beralih ke paket paralel jika biaya paket serial lebih dari 5 (yang mengacu pada unit biaya yang diabstraksi, bukan perkiraan waktu). Kami menyarankan Anda mengatur properti ini ke angka yang lebih tinggi.

Nilai default sesuai kembali ketika prosesor memiliki label harga tinggi, daya pemrosesan rendah, dan pemrosesan kueri lebih lambat dari sekarang. Prosesor saat ini jauh lebih cepat. Akibatnya, kueri yang relatif lebih kecil (misalnya, dengan ambang biaya 32) tidak akan mendapat banyak manfaat dari eksekusi paralel, terutama mengingat overhead yang terkait dengan koordinasi eksekusi paralel.

Dalam kebanyakan kasus, ambang biaya pengaturan paralelisme 50 adalah titik awal yang baik. Berikut adalah contoh cara mengkonfigurasi ambang biaya paralelisme:

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

Optimalkan beban kerja ad hoc

Aktifkan opsi optimasi untuk beban kerja ad hoc untuk meningkatkan efisiensi cache paket untuk beban kerja yang berisi banyak batch ad hoc sekali pakai. Saat Anda pertama kali menjalankan kueri ad hoc, mesin database menyimpan rintisan rencana yang dikompilasi alih-alih rencana eksekusi lengkap, sehingga menghemat ruang dalam cache paket. Jika Anda menjalankan batch ad hoc lagi, mesin database mengenali bahwa batch telah dijalankan sebelumnya dan mengganti rintisan rencana yang dikompilasi dengan paket lengkap yang dikompilasi dalam cache paket.

Untuk memeriksa apakah opsi ini diaktifkan, gunakan kueri:

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

Untuk informasi selengkapnya tentang mengoptimalkan beban kerja ad hoc, lihat dokumentasi Microsoft SQL Server.

Gunakan tanda jejak untuk meningkatkan kinerja

Pertimbangkan untuk menggunakan flag jejak SQL Server yang berlaku untuk lingkungan Anda untuk meningkatkan kinerja. Sebagai contoh:

- 4199: Mengaktifkan perubahan pengoptimal kueri (QO) yang dirilis di Pembaruan Kumulatif SQL Server (CU) dan Paket Layanan (SP).
- 8048: Mengonversi objek memori yang dipartisi NUMA menjadi objek memori yang dipartisi CPU.
- 9024: Mengonversi objek memori kumpulan log global ke objek memori yang dipartisi Numa.

Contoh berikut mengilustrasikan cara mengaktifkan dan menonaktifkan flag trace untuk SQL Server di Amazon EC2. Jika Anda mengalami masalah saat mengaktifkan penelusuran, pastikan Anda memiliki izin yang sesuai untuk akun tersebut.

Untuk mengaktifkan trace flag 4199, jalankan:

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

Untuk memeriksa status flag trace, jalankan:

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

Untuk menonaktifkan trace flag 4199, jalankan:

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

Untuk daftar lengkap flag jejak, lihat dokumentasi Microsoft SQL Server.

Instal tambalan terbaru

Dimulai dengan SQL Server 2017, Microsoft telah menghentikan rilis Paket Layanan (SP). Ini merilis Pembaruan Kumulatif (CU) dan pembaruan kritis (GDR) saja.

SP menyertakan perbaikan penting untuk SQL Server, jadi pastikan SP terbaru telah diinstal. Juga, jika memungkinkan, instal paket CU terbaru.

Untuk informasi tentang pembaruan SQL Server terbaru, lihat Pembaruan terbaru untuk Microsoft SQL Server di situs web Microsoft.

Tutup memori server maks untuk menghindari tekanan memori

Untuk alasan kinerja, SQL Server tidak melepaskan memori yang telah dialokasikan. Ketika SQL Server dimulai, perlahan-lahan mengambil memori yang ditentukan di bawah opsi min_server_memory, dan kemudian terus tumbuh hingga mencapai nilai yang ditentukan dalam opsi max_server_memory. (Untuk informasi selengkapnya tentang pengaturan ini, lihat Opsi konfigurasi memori server dalam dokumentasi SQL Server.)

Memori SQL Server memiliki dua komponen: buffer pool dan non-buffer pool (juga disebut memory to leave atau MTL). Nilai opsi max_server_memory menentukan ukuran kumpulan buffer SQL Server, yang terdiri dari cache buffer, cache prosedur, cache rencana, struktur buff, dan cache lainnya.

Dimulai dengan SQL Server 2012, akun min_server_memory dan max_server_memory untuk semua alokasi memori untuk semua cache, termasuk,,,,,, dan. `SQLGENERAL` `SQLBUFFERPOOL` `SQLQUERYCOMPILE` `SQLQUERYPLAN` `SQLQUERYEXEC` `SQLOPTIMIZER` `SQLCLR` Untuk daftar lengkap pegawai memori di bawah max_server_memory, lihat sys.dm_os_memory_clerks dalam dokumentasi Microsoft SQL Server.

Untuk memeriksa nilai max_server_memory saat ini, gunakan perintah:

```
$ sp_configure 'max_server_memory'
```

Kami menyarankan Anda membatasi max_server_memory pada nilai yang tidak menyebabkan tekanan memori di seluruh sistem. Tidak ada rumus universal yang berlaku untuk semua lingkungan, tetapi kami telah memberikan beberapa pedoman di bagian ini. max_server_memory adalah opsi dinamis, sehingga dapat diubah pada waktu berjalan.

Sebagai titik awal, Anda dapat menentukan max_server_memory sebagai berikut:

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

di mana:

- Memori untuk sistem operasi adalah 1-4 GB.
- MTL (memori untuk meninggalkan) mencakup ukuran tumpukan, yaitu 2 MB pada mesin 64-bit per thread pekerja dan dapat dihitung sebagai berikut: `MTL = stack_size * max_worker_threads`

Atau, Anda dapat menggunakan:

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

dimana jumlah dasar memori RAM ditentukan sebagai berikut:

- Jika RAM di server antara 4 GB dan 16 GB, sisakan 1 GB per 4 GB RAM. Misalnya, untuk server dengan 16 GB, sisakan 4 GB.
- Jika RAM di server lebih dari 16 GB, sisakan 1 GB per 4 GB RAM hingga 16 GB, dan 1 GB per 8 GB RAM di atas 16 GB.

Misalnya, jika server memiliki 256 GB RAM, perhitungannya adalah:

- 1 GB untuk OS
- Hingga 16 GB RAM: 16/4 = 4 GB
- Sisa RAM di atas 16 GB: (256-16) /8 = 30
- Total RAM untuk pergi: 1 + 4 + 30 = 35 GB

- max_server_memory: 256 - 35 = 221 GB

Setelah konfigurasi awal, pantau memori yang dapat Anda bebaskan selama durasi beban kerja biasa untuk menentukan apakah Anda perlu menambah atau mengurangi memori yang dialokasikan ke SQL Server.

> **Note**
>
> Windows memberi sinyal notifikasi sumber daya memori rendah pada 96 MB, jadi Anda menginginkan buffer, tetapi Anda dapat mengatur Mbytes yang Tersedia menjadi di atas 1 GB pada server yang lebih besar dengan RAM 256 GB atau lebih tinggi.

Untuk informasi tambahan, lihat Panduan Arsitektur Manajemen Memori di dokumentasi Microsoft SQL Server.

Gunakan tingkat kompatibilitas database tertinggi

Periksa untuk memastikan bahwa Anda menggunakan tingkat kompatibilitas database saat ini untuk memanfaatkan peningkatan terbaru di SQL Server. Ini penting untuk diperiksa karena ketika Anda mengembalikan database dari versi yang lebih rendah ke versi yang lebih tinggi, itu akan menjaga tingkat kompatibilitas versi yang lebih rendah. Beberapa peningkatan database terbaru hanya efektif ketika Anda mengatur kompatibilitas database ke tingkat terbaru yang tersedia untuk versi mesin yang Anda instal.

Untuk memeriksa kompatibilitas database saat ini, gunakan:

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

Untuk informasi selengkapnya tentang tingkat kompatibilitas database, lihat dokumentasi Microsoft SQL Server.

Kontrol jumlah VLF

Pra-alokasikan ukuran maksimum data dan file log. Untuk kinerja yang lebih baik, kendalikan jumlah file log virtual (VLF) dengan mengalokasikan ruang terlebih dahulu dan mengoreksi pengaturan pertumbuhan otomatis (autogrow) untuk file log.

Biasanya, faktor autogrow 8 GB bekerja dengan baik di sebagian besar lingkungan produksi. Pertimbangkan untuk menumbuhkan file log transaksi dalam potongan 8-GB. Jumlah VLF yang lebih tinggi dapat memperpanjang waktu pencadangan dan pemulihan untuk database, dan dapat menyebabkan masalah kinerja dengan operasi apa pun (misalnya, replikasi) yang memerlukan melalui file log.

Untuk informasi lebih lanjut tentang pembuatan dan algoritma pertumbuhan VLF, lihat blog SQLSkills.

Periksa pengaturan autogrowth database

Setiap transaksi yang membutuhkan data atau file log untuk tumbuh mencakup waktu yang dibutuhkan oleh operasi pertumbuhan file. File tumbuh dengan ukuran kenaikan yang ditentukan oleh opsi FILEGROWTH. Anda dapat mencari peristiwa pertumbuhan file di jejak profiler SQL Server. Jika pertumbuhan file membutuhkan waktu lama, Anda mungkin melihat jenis tunggu seperti`ASYNC_IO_COMPLETION`, yang terjadi ketika pemrosesan data sangat lambat. Jenis tunggu seperti itu tidak hanya memengaruhi kinerja tetapi juga dapat mengakibatkan batas waktu transaksi. Jika transaksi itu menahan kunci pada sumber daya yang dicari oleh transaksi lain, batas waktu akan menyebabkan masalah pemblokiran server yang parah.

Untuk alasan ini, kami menyarankan Anda mengonfigurasi pengaturan pertumbuhan otomatis dengan sangat hati-hati. Juga perlu diingat bahwa:

- Pertumbuhan file adalah salah satu operasi termahal di SQL Server.
- Pertumbuhan otomatis yang sering dalam potongan kecil dapat menyebabkan fragmentasi disk.
- Pertumbuhan otomatis yang sering dalam file log menghasilkan sejumlah besar file log virtual (VLF) dan memengaruhi kinerja, seperti yang dibahas di bagian sebelumnya.

Semua alasan ini dapat menyebabkan startup basis data yang lambat dan peningkatan waktu pencadangan dan pemulihan.

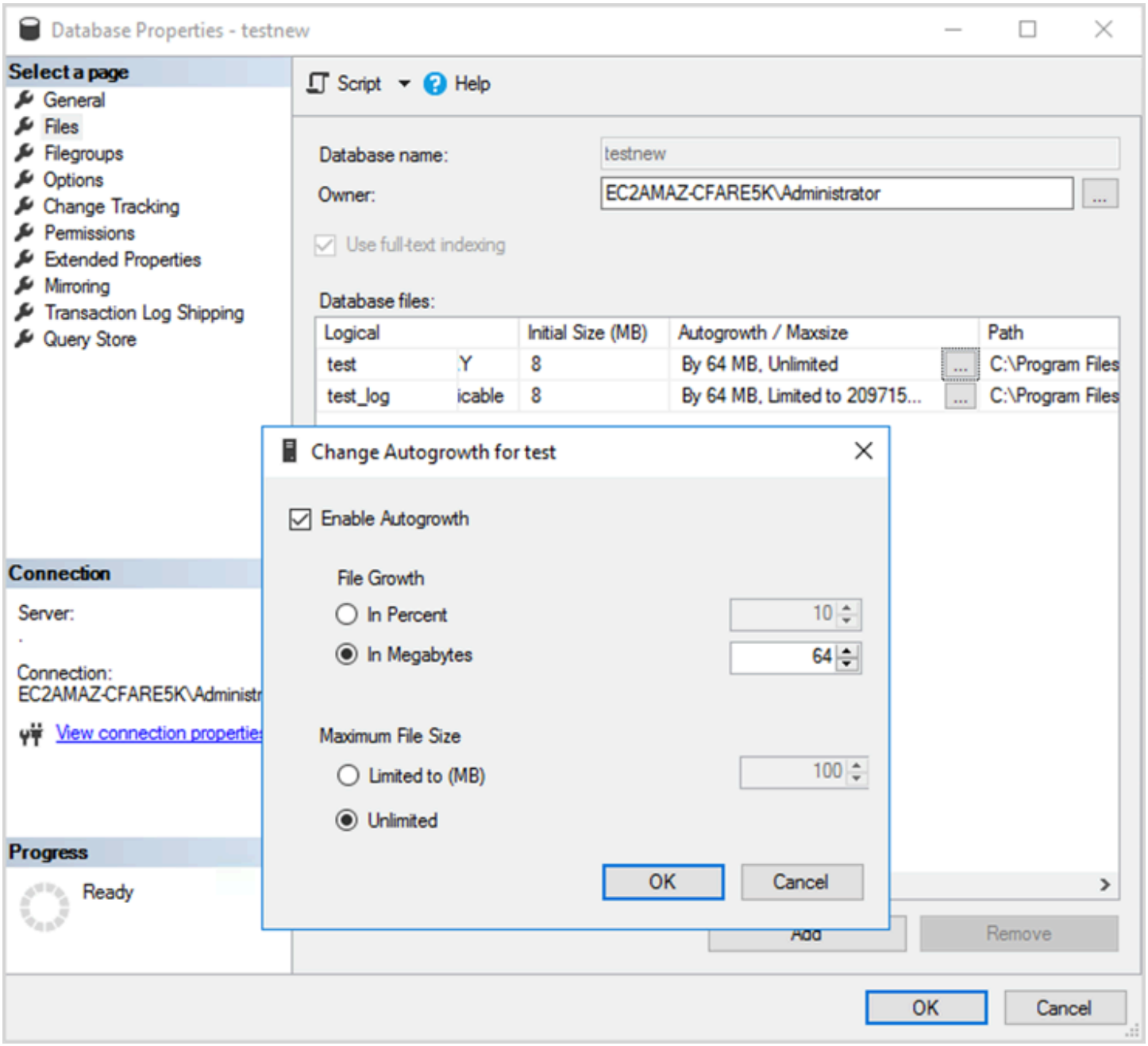
Idealnya, Anda harus melakukan pra-pertumbuhan file secara proaktif, berdasarkan pemantauan rutin. Pilih dengan hati-hati antara pengaturan autogrowth sebagai persentase atau sebagai nilai statis (dalam MB). Biasanya, mengatur pertumbuhan otomatis ke seperdelapan ukuran file adalah titik awal yang baik, tetapi ini mungkin bukan pilihan yang tepat. (Misalnya, persentase ini akan terlalu tinggi jika file data Anda berukuran beberapa TB.)

Dalam kebanyakan kasus, nilai autogrowth 1024 MB berfungsi dengan baik untuk file data di sebagian besar database besar. Untuk file log, 512 MB adalah titik awal yang baik. Untuk ukuran

kontingensi, kami sangat menyarankan Anda menetapkan nilai pertumbuhan otomatis, tetapi menumbuhkan file secara manual selama beberapa bulan berdasarkan tren sebelumnya.

> **Note**
>
> Pengaturan autogrowth harus menjadi ukuran kontingensi, jadi Anda harus mengaturnya setelah Anda mengalokasikan penyimpanan terlebih dahulu ke file.

Anda dapat mengubah pengaturan autogrowth dengan menggunakan SQL Server Management Studio (SSMS) atau Transact-SQL. Ilustrasi layar berikut menunjukkan pengaturan autogrowth di SSMS.



Saat Anda menggunakan opsi FILEGROWTH untuk file data dan log, pilih dengan hati-hati antara mengaturnya sebagai persentase atau sebagai nilai statis (dalam MB). Menetapkan persentase menghasilkan pertumbuhan file yang terus meningkat, jadi Anda mungkin lebih suka menggunakan ukuran statis untuk kontrol yang lebih baik atas rasio pertumbuhan.

- Dalam versi sebelum SQL Server 2022 (16.x), log transaksi tidak dapat menggunakan inisialisasi file instan, jadi waktu pertumbuhan log yang diperpanjang sangat penting.

- Dimulai dengan SQL Server 2022 (16.x, semua edisi), inisialisasi file instan dapat menguntungkan peristiwa pertumbuhan log transaksi hingga 64 MB. Peningkatan ukuran autogrowth default untuk database baru adalah 64 MB. Peristiwa pertumbuhan otomatis file log transaksi yang lebih besar dari 64 MB tidak dapat memperoleh manfaat dari inisialisasi file instan.

Mengkonfigurasi grup ketersediaan Selalu Aktif

Jika Anda menggunakan pustaka klien asli untuk SQL Server versi 2012 dan yang lebih baru, dan pustaka .NET Framework 4.5, Anda dapat menggunakan MultiSubnetFailoverparameter untuk mengubah perilaku sambungan. Kami menyarankan Anda mengatur parameter ini ke`TRUE`. Ini akan memungkinkan failover lebih cepat dengan grup ketersediaan Always On.

> **Note**
>
> Jika Anda memiliki aplikasi lama yang tidak dapat menggunakan MultiSubnetFailoverparameter, Anda dapat menempatkan Network Load Balancer di depan instance SQL Server Anda. Penyeimbang menggunakan pemeriksaan kesehatan yang menentukan database SQL Server yang aktif dan mengirimkan lalu lintas ke instance yang saat ini meng-host database tersebut. Penyeimbang beban mencakup satu atau beberapa Availability Zone. Anda dapat menggunakan port khusus seperti 59999 untuk pemeriksaan kesehatan, dan kemudian memodifikasi parameter grup klaster untuk merespons port tersebut. Hal ini memungkinkan Anda untuk mengurangi SQL Server failover waktu untuk sekitar satu menit tanpa menggunakan MultiSubnetFailoverparameter. Untuk petunjuk terperinci, lihat posting blog Kurangi waktu failover untuk SQL Server di instans Amazon EC2 menggunakan Network Load Balancer.

Dua pengaturan memengaruhi cara listener grup ketersediaan terdaftar dengan DNS: RegisterAllProvidersIP dan HostRecordTTL.

Setel RegisterAllProviders IP ke true saat menggunakan grup ketersediaan Always On

Kami menyarankan Anda mengatur RegisterAllProvidersIP ke`1` (true). Ketika listener grup ketersediaan dibuat dengan RegisterAllProvidersIP diatur ke`1`, semua alamat IP untuk pendengar tersebut terdaftar di DNS. Ketika RegisterAllProvidersIP diatur ke`0` (false), hanya satu IP aktif yang terdaftar.

Dalam kasus failover, ketika replika utama bergerak dari satu subnet ke subnet lainnya, alamat IP lama tidak terdaftar, dan alamat IP baru terdaftar. DNS diperbarui dengan IP baru saat listener grup ketersediaan online. Namun, sistem klien tidak akan menyelesaikan nama pendengar ke alamat IP baru sampai entri yang saat ini di-cache berakhir.

Setel HostRecord TTL ke 60 atau kurang saat menggunakan grup ketersediaan Always On

Pengaturan HostRecordTTL mengontrol Waktu untuk Live (TTL) untuk entri DNS yang disimpan. Nilai default adalah 1200 detik. Kami menyarankan Anda mengubah HostRecordTTL ke pengaturan yang jauh lebih rendah (60 detik atau kurang). Hal ini menyebabkan nilai cache berakhir lebih cepat, sehingga dalam kasus failover, sistem klien dapat menyelesaikan IP baru lebih cepat.

Nonaktifkan failback otomatis untuk grup klaster Always On

Verifikasi bahwa kegagalan otomatis dinonaktifkan untuk grup ketersediaan Always On di Windows Cluster Manager.

Mengkonfigurasi backup

Seperti yang dibahas dalamOptimalkan tata letak disk atau distribusi filebagian, kami sarankan Anda mengirim backup SQL Server asli Anda ke drive terpisah. Juga pertimbangkan untuk mengambil snapshot terjadwal volume EBS tempat file cadangan berada.

Meningkatkan optimasi database

Bagian ini menyediakan praktik terbaik untuk meningkatkan kinerja saat bekerja dengan pengoptimal kueri SQL Server. Ini membahas bagaimana membangun kembali indeks dan memperbarui statistik tabel secara teratur dapat membantu mengoptimalkan rencana eksekusi. Bagian berikut menyediakan kiat konfigurasi dan praktik terbaik.

Topik

Indeks membangun ulang

Untuk query optimizer untuk menghasilkan rencana query terbaik dan menggunakan indeks yang tepat, indeks tidak boleh terfragmentasi. Indeks menjadi terfragmentasi dari waktu ke waktu berdasarkan update, insert, atau delete rate. Pastikan bahwa tabel diindeks ulang secara teratur. Frekuensi membangun ulang tergantung pada tingkat di mana database menangani operasi data manipulation language (DL).

Titik awal yang baik adalah membangun kembali indeks yang terfragmentasi lebih dari 30%, dan mengatur ulang indeks yang terfragmentasi kurang dari 30%. Nilai 30% bekerja di sebagian besar kasus penggunaan, tetapi jika Anda masih melihat rencana kueri yang buruk karena indeks yang tidak terpakai, Anda mungkin perlu meninjau kembali persentase ini.

Gunakan kueri seperti berikut ini untuk memeriksa fragmentasi:

```
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
```

Kami merekomendasikan agar Anda membuat pekerjaan pemeliharaan untuk membangun ulang indeks secara teratur.

Statistik pembaruan

Seperti indeks terfragmentasi, jika optimizer tidak memiliki informasi terbaru tentang distribusi nilai kunci (statistik) kolom tabel, itu tidak dapat menghasilkan rencana eksekusi optimal. Sebaiknya Anda memperbarui statistik untuk semua tabel secara teratur. Frekuensi pembaruan tergantung pada tingkat di mana database menangani operasi DML-nya, tetapi biasanya dijalankan dua kali seminggu selama jam sibuk. Namun, hindari memperbarui statistik pada hari-hari ketika Anda membangun kembali indeks. Untuk informasi selengkapnya mengenai statistik pembaruan, lihatDokumentasi Microsoft SQL Server.

Untuk optimasi database, sebaiknya gunakan skrip pemeliharaan indeks dan statistik. Sebagai contoh, lihatSQL Server indeks dan statistik pemeliharaan skripdisediakan di situs web solusi pemeliharaan SQL Server.

Mengoptimalkan penerapan SQL Server di Amazon EC2 denganAWS Launch Wizard

AWSLaunch Wizard adalah metode utama untuk penerapan instans tunggal SQL Server dan ketersediaan tinggi (HA) di Amazon EC2. Penyebaran Launch Wizard didasarkan pada AWSWell-Architected Framework, dan dioptimalkan untuk keamanan, keandalan, efisiensi kinerja, dan penghematan biaya.

Launch Wizard menyederhanakan penyebaran SQL Server Anda dan juga membuatnya lebih mudah untuk mengkonfigurasi SQL Server. Fitur meliputi:

- PemilihanAWS sumber daya otomatis - Launch Wizard dapat merekomendasikan jenis instans optimal berdasarkan CPU virtual (vCPU), memori, dan persyaratan jaringan Anda. Ini juga dapat merekomendasikan jenis volume berdasarkan drive penyimpanan dan throughput.
- Pemantauan sekali klik — Launch Wizard terintegrasi dengan Amazon CloudWatch Application Insights untuk menyiapkan pemantauan penerapan SQL Server HA aktifAWS. Saat Anda memilih opsi ini, Wawasan Aplikasi secara otomatis menyiapkan metrik, log, dan alarm yang relevan CloudWatch, dan mulai memantau beban kerja yang baru diterapkan.
- Grup sumber daya aplikasi untuk mudah ditemukan — Launch Wizard membuat grup sumber daya untuk semuaAWS sumber daya yang dibuat untuk aplikasi SQL Server Anda. Anda dapat mengelola, menambal, dan memelihara aplikasi SQL Server Anda dariAWS Systems Manager konsol.

Launch Wizard memberi Anda templatAWS CloudFormation kode yang dapat digunakan kembali. Template ini dapat berfungsi sebagai dasar untuk penerapan aplikasi Anda berikutnya. Untuk mempelajari selengkapnya, lihat ringkasanAWS Launch Wizard dan panduan pengguna.

Langkah selanjutnya

Panduan ini membahas beberapa praktik terbaik untuk mengonfigurasi dan menjalankan beban kerja Microsoft SQL Server di Amazon EC2. Mengikuti pedoman ini dalam fase perencanaan dan implementasi proses migrasi akan membantu Anda membangun server yang stabil di lingkungan produksi.

Untuk informasi selengkapnya tentang tugas konfigurasi ini, lihat tautan yang disediakan di setiap bagian dan kunjungi halaman web yang tercantum dalamSumber daya tambahanBagian.

Sumber daya tambahan

Strategi, panduan, dan pola terkait

- Strategi migrasi untuk database relasional
- Pola SQL Server:
  - Pola semua
  - Pola rehost(memigrasi SQL Server ke Amazon EC2)
  - Pola replatform(Migrasi SQL Server ke Amazon RDS for SQL Server)
  - Pola arsitek(memigrasi SQL Server ke open-source danAWSDatabase cloud-native)
- AWSSitus Prescriptive Guidance

AWSsumber daya

- AWSdokumentasi
- AWSReferensi umum
- AWSGlosarium

AWSjasa

- EBS Amazon
- Amazon EC2

Sumber daya lainnya

- Volume EBS tidak diinisialisasi di Windows Server 2016 dan AMI yang lebih baru
- Cara memindahkan tempdb Micrsoft SQL Server ke disk instance/sementara di Amazon EC2
- Menjalankan perintah pada instance Windows Anda saat peluncuran
- Stripe Windows Disk Ephemeral pada Peluncuran
- Benchmarking SQL Server dengan HammerDB
- Berapa banyak memori yang sebenarnya dibutuhkan SQL Server saya?
- SQL Server Tunggu Statistik (atau tolong beritahu saya di mana sakit...)
- Waktu Koneksi di Grup Ketersediaan Multi-subnet

- Rencanakan cache dan optimalisasi untuk beban kerja adhoc
- RAM, memori virtual, pagefile, dan manajemen memori di Windows
- Cara menentukan ukuran file halaman yang sesuai untuk versi Windows 64-bit

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Informasi yang dikoreksi	Memperbaiki informasi tentang ambang biaya properti paralelisme. Nilainya diukur dengan satuan biaya, bukan waktu.	Desember 4, 2023
Panduan yang diperbarui	Memperbarui bagian tentang pengaturan ukuran unit alokasi NTFS, mengunci halaman dalam memori, menggunakan fitur pembongkaran tugas, menggunakan striping, mengubah ambang biaya paralelisme, menggunakan flag jejak, dan menggunakan pengaturan autogrowth database.	8 Agustus 2023
Menambahkan panduan	Menambahkan informasi tentang penggunaan Network Load Balancer untuk aplikasi lama yang tidak dapat menggunakan parameter. MultiSubnetFailover	11 November 2022
Kode tetap	Memperbaiki PowerShell kode di bagian tentang menginisi alisasi penyimpanan instance.	Juni 27, 2022

Ditambahkan bagian baru	Menambahkan informasi tentang AWSLaunch Wizard untuk SQL Server.	18 Agustus 2021
Publikasi awal	—	21 Juli 2020

AWS Glosarium Panduan Preskriptif

Berikut ini adalah istilah yang umum digunakan dalam strategi, panduan, dan pola yang disediakan oleh Panduan AWS Preskriptif. Untuk menyarankan entri, silakan gunakan tautan Berikan umpan balik di akhir glosarium.

Nomor

7 Rs

Tujuh strategi migrasi umum untuk memindahkan aplikasi ke cloud. Strategi ini dibangun di atas 5 Rs yang diidentifikasi Gartner pada tahun 2011 dan terdiri dari yang berikut:

- Refactor/Re-Architect — Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di Cloud. AWS
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com.
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instans EC2 di Cloud. AWS
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Skenario migrasi ini khusus untuk VMware Cloud on AWS, yang mendukung kompatibilitas mesin virtual (VM) dan portabilitas beban kerja antara lingkungan lokal Anda dan. AWS Anda dapat menggunakan teknologi VMware Cloud Foundation dari pusat data lokal saat memigrasikan infrastruktur ke VMware Cloud. AWS Contoh: Pindahkan hypervisor yang menghosting database Oracle Anda ke VMware Cloud on. AWS
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu

sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

- Pensiun — Menonaktifkan atau menghapus aplikasi yang tidak lagi diperlukan di lingkungan sumber Anda.

A

ABAC

Lihat kontrol akses berbasis atribut.

layanan abstrak

Lihat layanan terkelola.

ASAM

Lihat atomisitas, konsistensi, isolasi, daya tahan.

migrasi aktif-aktif

Metode migrasi database di mana basis data sumber dan target tetap sinkron (dengan menggunakan alat replikasi dua arah atau operasi penulisan ganda), dan kedua database menangani transaksi dari menghubungkan aplikasi selama migrasi. Metode ini mendukung migrasi dalam batch kecil yang terkontrol alih-alih memerlukan pemotongan satu kali. Ini lebih fleksibel tetapi membutuhkan lebih banyak pekerjaan daripada migrasi aktif-pasif.

migrasi aktif-pasif

Metode migrasi database di mana database sumber dan target disimpan dalam sinkron, tetapi hanya database sumber yang menangani transaksi dari menghubungkan aplikasi sementara data direplikasi ke database target. Basis data target tidak menerima transaksi apa pun selama migrasi.

fungsi agregat

Fungsi SQL yang beroperasi pada sekelompok baris dan menghitung nilai pengembalian tunggal untuk grup. Contoh fungsi agregat meliputi `SUM` dan`MAX`.

AI

Lihat kecerdasan buatan.

AIOps

Lihat operasi kecerdasan buatan.

anonimisasi

Proses menghapus informasi pribadi secara permanen dalam kumpulan data. Anonimisasi dapat membantu melindungi privasi pribadi. Data anonim tidak lagi dianggap sebagai data pribadi.

anti-pola

Solusi yang sering digunakan untuk masalah berulang di mana solusinya kontra-produktif, tidak efektif, atau kurang efektif daripada alternatif.

kontrol aplikasi

Pendekatan keamanan yang memungkinkan penggunaan hanya aplikasi yang disetujui untuk membantu melindungi sistem dari malware.

portofolio aplikasi

Kumpulan informasi rinci tentang setiap aplikasi yang digunakan oleh organisasi, termasuk biaya untuk membangun dan memelihara aplikasi, dan nilai bisnisnya. Informasi ini adalah kunci untuk penemuan portofolio dan proses analisis dan membantu mengidentifikasi dan memprioritaskan aplikasi yang akan dimigrasi, dimodernisasi, dan dioptimalkan.

kecerdasan buatan (AI)

Bidang ilmu komputer yang didedikasikan untuk menggunakan teknologi komputasi untuk melakukan fungsi kognitif yang biasanya terkait dengan manusia, seperti belajar, memecahkan masalah, dan mengenali pola. Untuk informasi lebih lanjut, lihat Apa itu Kecerdasan Buatan?

operasi kecerdasan buatan (AIOps)

Proses menggunakan teknik pembelajaran mesin untuk memecahkan masalah operasional, mengurangi insiden operasional dan intervensi manusia, dan meningkatkan kualitas layanan. Untuk informasi selengkapnya tentang cara AIOps digunakan dalam strategi AWS migrasi, lihat panduan integrasi operasi.

enkripsi asimetris

Algoritma enkripsi yang menggunakan sepasang kunci, kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Anda dapat berbagi kunci publik karena tidak digunakan untuk dekripsi, tetapi akses ke kunci pribadi harus sangat dibatasi.

atomisitas, konsistensi, isolasi, daya tahan (ACID)

Satu set properti perangkat lunak yang menjamin validitas data dan keandalan operasional database, bahkan dalam kasus kesalahan, kegagalan daya, atau masalah lainnya.

kontrol akses berbasis atribut (ABAC)

Praktik membuat izin berbutir halus berdasarkan atribut pengguna, seperti departemen, peran pekerjaan, dan nama tim. Untuk informasi selengkapnya, lihat ABAC untuk AWS dokumentasi AWS Identity and Access Management (IAM).

sumber data otoritatif

Lokasi di mana Anda menyimpan versi utama data, yang dianggap sebagai sumber informasi yang paling dapat diandalkan. Anda dapat menyalin data dari sumber data otoritatif ke lokasi lain untuk tujuan memproses atau memodifikasi data, seperti menganonimkan, menyunting, atau membuat nama samaran.

Zona Ketersediaan

Lokasi berbeda di dalam Wilayah AWS yang terisolasi dari kegagalan di Availability Zone lainnya dan menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama.

AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat situs web AWS CAF dan whitepaper AWS CAF.

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

Alat yang mengevaluasi beban kerja migrasi database, merekomendasikan strategi migrasi, dan memberikan perkiraan kerja. AWS WQF disertakan dengan AWS Schema Conversion Tool ()AWS SCT. Ini menganalisis skema database dan objek kode, kode aplikasi, dependensi, dan karakteristik kinerja, dan memberikan laporan penilaian.

B

bot buruk

Bot yang dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

BCP

Lihat perencanaan kontinuitas bisnis.

grafik perilaku

Pandangan interaktif yang terpadu tentang perilaku dan interaksi sumber daya dari waktu ke waktu. Anda dapat menggunakan grafik perilaku dengan Amazon Detective untuk memeriksa upaya logon yang gagal, panggilan API yang mencurigakan, dan tindakan serupa. Untuk informasi selengkapnya, lihat Data dalam grafik perilaku di dokumentasi Detektif.

sistem big-endian

Sistem yang menyimpan byte paling signifikan terlebih dahulu. Lihat juga endianness.

klasifikasi biner

Sebuah proses yang memprediksi hasil biner (salah satu dari dua kelas yang mungkin). Misalnya, model ML Anda mungkin perlu memprediksi masalah seperti “Apakah email ini spam atau bukan spam?” atau “Apakah produk ini buku atau mobil?”

filter mekar

Struktur data probabilistik dan efisien memori yang digunakan untuk menguji apakah suatu elemen adalah anggota dari suatu himpunan.

deployment biru/hijau

Strategi penyebaran tempat Anda membuat dua lingkungan yang terpisah namun identik. Anda menjalankan versi aplikasi saat ini di satu lingkungan (biru) dan versi aplikasi baru di lingkungan lain (hijau). Strategi ini membantu Anda dengan cepat memutar kembali dengan dampak minimal.

bot

Aplikasi perangkat lunak yang menjalankan tugas otomatis melalui internet dan mensimulasikan aktivitas atau interaksi manusia. Beberapa bot berguna atau bermanfaat, seperti perayap web yang mengindeks informasi di internet. Beberapa bot lain, yang dikenal sebagai bot buruk, dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

botnet

Jaringan bot yang terinfeksi oleh malware dan berada di bawah kendali satu pihak, yang dikenal sebagai bot herder atau operator bot. Botnet adalah mekanisme paling terkenal untuk skala bot dan dampaknya.

cabang

Area berisi repositori kode. Cabang pertama yang dibuat dalam repositori adalah cabang utama. Anda dapat membuat cabang baru dari cabang yang ada, dan Anda kemudian dapat mengembangkan fitur atau memperbaiki bug di cabang baru. Cabang yang Anda buat untuk membangun fitur biasanya disebut sebagai cabang fitur. Saat fitur siap dirilis, Anda menggabungkan cabang fitur kembali ke cabang utama. Untuk informasi selengkapnya, lihat Tentang cabang (GitHub dokumentasi).

akses break-glass

Dalam keadaan luar biasa dan melalui proses yang disetujui, cara cepat bagi pengguna untuk mendapatkan akses ke Akun AWS yang biasanya tidak memiliki izin untuk mengaksesnya. Untuk informasi lebih lanjut, lihat indikator Implementasikan prosedur break-glass dalam panduan Well-Architected AWS .

strategi brownfield

Infrastruktur yang ada di lingkungan Anda. Saat mengadopsi strategi brownfield untuk arsitektur sistem, Anda merancang arsitektur di sekitar kendala sistem dan infrastruktur saat ini. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

Apa yang dilakukan bisnis untuk menghasilkan nilai (misalnya, penjualan, layanan pelanggan, atau pemasaran). Arsitektur layanan mikro dan keputusan pengembangan dapat didorong oleh kemampuan bisnis. Untuk informasi selengkapnya, lihat bagian Terorganisir di sekitar kemampuan bisnis dari Menjalankan layanan mikro kontainer di whitepaper. AWS

perencanaan kelangsungan bisnis (BCP)

Rencana yang membahas dampak potensial dari peristiwa yang mengganggu, seperti migrasi skala besar, pada operasi dan memungkinkan bisnis untuk melanjutkan operasi dengan cepat.

C

KAFE

Lihat Kerangka Adopsi AWS Cloud.

penyebaran kenari

Rilis versi yang lambat dan bertahap untuk pengguna akhir. Ketika Anda yakin, Anda menyebarkan versi baru dan mengganti versi saat ini secara keseluruhan.

CCoE

Lihat Cloud Center of Excellence.

CDC

Lihat mengubah pengambilan data.

ubah pengambilan data (CDC)

Proses melacak perubahan ke sumber data, seperti tabel database, dan merekam metadata tentang perubahan tersebut. Anda dapat menggunakan CDC untuk berbagai tujuan, seperti mengaudit atau mereplikasi perubahan dalam sistem target untuk mempertahankan sinkronisasi.

rekayasa kekacauan

Sengaja memperkenalkan kegagalan atau peristiwa yang mengganggu untuk menguji ketahanan sistem. Anda dapat menggunakan AWS Fault Injection Service (AWS FIS) untuk melakukan eksperimen yang menekankan AWS beban kerja Anda dan mengevaluasi responsnya.

CI/CD

Lihat integrasi berkelanjutan dan pengiriman berkelanjutan.

klasifikasi

Proses kategorisasi yang membantu menghasilkan prediksi. Model ML untuk masalah klasifikasi memprediksi nilai diskrit. Nilai diskrit selalu berbeda satu sama lain. Misalnya, model mungkin perlu mengevaluasi apakah ada mobil dalam gambar atau tidak.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Cloud Center of Excellence (CCoE)

Tim multi-disiplin yang mendorong upaya adopsi cloud di seluruh organisasi, termasuk mengembangkan praktik terbaik cloud, memobilisasi sumber daya, menetapkan jadwal migrasi, dan memimpin organisasi melalui transformasi skala besar. Untuk informasi selengkapnya, lihat posting CCoE di Blog Strategi AWS Cloud Enterprise.

komputasi cloud

Teknologi cloud yang biasanya digunakan untuk penyimpanan data jarak jauh dan manajemen perangkat IoT. Cloud computing umumnya terhubung ke teknologi edge computing.

model operasi cloud

Dalam organisasi TI, model operasi yang digunakan untuk membangun, mematangkan, dan mengoptimalkan satu atau lebih lingkungan cloud. Untuk informasi selengkapnya, lihat Membangun Model Operasi Cloud Anda.

tahap adopsi cloud

Empat fase yang biasanya dilalui organisasi ketika mereka bermigrasi ke AWS Cloud:

- Proyek — Menjalankan beberapa proyek terkait cloud untuk bukti konsep dan tujuan pembelajaran
- Foundation — Melakukan investasi dasar untuk meningkatkan adopsi cloud Anda (misalnya, membuat landing zone, mendefinisikan CCoE, membuat model operasi)
- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog The Journey Toward Cloud-First & the Stages of Adoption di blog AWS Cloud Enterprise Strategy. Untuk informasi tentang bagaimana kaitannya dengan strategi AWS migrasi, lihat panduan kesiapan migrasi.

CMDB

Lihat database manajemen konfigurasi.

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau. AWS CodeCommit Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori

dikhususkan untuk satu bagian fungsionalitas. Pipa CI/CD tunggal dapat menggunakan beberapa repositori.

cache dingin

Cache buffer yang kosong, tidak terisi dengan baik, atau berisi data basi atau tidak relevan. Ini mempengaruhi kinerja karena instance database harus membaca dari memori utama atau disk, yang lebih lambat daripada membaca dari cache buffer.

data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat atau kelas penyimpanan yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer (CV)

Bidang AI yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, AWS Panorama menawarkan perangkat yang menambahkan CV ke jaringan kamera lokal, dan Amazon SageMaker menyediakan algoritme pemrosesan gambar untuk CV.

konfigurasi drift

Untuk beban kerja, konfigurasi berubah dari status yang diharapkan. Ini dapat menyebabkan beban kerja menjadi tidak patuh, dan biasanya bertahap dan tidak disengaja.

database manajemen konfigurasi (CMDB)

Repositori yang menyimpan dan mengelola informasi tentang database dan lingkungan TI, termasuk komponen perangkat keras dan perangkat lunak dan konfigurasinya. Anda biasanya menggunakan data dari CMDB dalam penemuan portofolio dan tahap analisis migrasi.

paket kesesuaian

Kumpulan AWS Config aturan dan tindakan remediasi yang dapat Anda kumpulkan untuk menyesuaikan kepatuhan dan pemeriksaan keamanan Anda. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal di Akun AWS dan Region, atau di seluruh organisasi, dengan menggunakan templat YAMM. Untuk informasi selengkapnya, lihat Paket kesesuaian dalam dokumentasi. AWS Config

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD umumnya digambarkan sebagai pipa. CI/CD dapat membantu

Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat Manfaat pengiriman berkelanjutan. CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat Continuous Delivery vs Continuous Deployment.

CV

Lihat visi komputer.

D

data saat istirahat

Data yang stasioner di jaringan Anda, seperti data yang ada di penyimpanan.

klasifikasi data

Proses untuk mengidentifikasi dan mengkategorikan data dalam jaringan Anda berdasarkan kekritisan dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat Klasifikasi data.

penyimpangan data

Variasi yang berarti antara data produksi dan data yang digunakan untuk melatih model ML, atau perubahan yang berarti dalam data input dari waktu ke waktu. Penyimpangan data dapat mengurangi kualitas, akurasi, dan keadilan keseluruhan dalam prediksi model ML.

data dalam transit

Data yang aktif bergerak melalui jaringan Anda, seperti antara sumber daya jaringan.

jala data

Kerangka arsitektur yang menyediakan kepemilikan data terdistribusi dan terdesentralisasi dengan manajemen dan tata kelola terpusat.

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data AWS Cloud dapat mengurangi risiko privasi, biaya, dan jejak karbon analitik Anda.

perimeter data

Satu set pagar pembatas pencegahan di AWS lingkungan Anda yang membantu memastikan bahwa hanya identitas tepercaya yang mengakses sumber daya tepercaya dari jaringan yang diharapkan. Untuk informasi selengkapnya, lihat Membangun perimeter data pada AWS.

prapemrosesan data

Untuk mengubah data mentah menjadi format yang mudah diuraikan oleh model ML Anda. Preprocessing data dapat berarti menghapus kolom atau baris tertentu dan menangani nilai yang hilang, tidak konsisten, atau duplikat.

asal data

Proses melacak asal dan riwayat data sepanjang siklus hidupnya, seperti bagaimana data dihasilkan, ditransmisikan, dan disimpan.

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

Sistem manajemen data yang mendukung intelijen bisnis, seperti analitik. Gudang data biasanya berisi sejumlah besar data historis, dan biasanya digunakan untuk kueri dan analisis.

bahasa definisi database (DDL)

Pernyataan atau perintah untuk membuat atau memodifikasi struktur tabel dan objek dalam database.

bahasa manipulasi basis data (DHTML)

Pernyataan atau perintah untuk memodifikasi (memasukkan, memperbarui, dan menghapus) informasi dalam database.

DDL

Lihat bahasa definisi database.

ansambel yang dalam

Untuk menggabungkan beberapa model pembelajaran mendalam untuk prediksi. Anda dapat menggunakan ansambel dalam untuk mendapatkan prediksi yang lebih akurat atau untuk memperkirakan ketidakpastian dalam prediksi.

pembelajaran mendalam

Subbidang ML yang menggunakan beberapa lapisan jaringan saraf tiruan untuk mengidentifikasi pemetaan antara data input dan variabel target yang diinginkan.

defense-in-depth

Pendekatan keamanan informasi di mana serangkaian mekanisme dan kontrol keamanan dilapisi dengan cermat di seluruh jaringan komputer untuk melindungi kerahasiaan, integritas, dan ketersediaan jaringan dan data di dalamnya. Saat Anda mengadopsi strategi ini AWS, Anda menambahkan beberapa kontrol pada lapisan AWS Organizations struktur yang berbeda untuk membantu mengamankan sumber daya. Misalnya, defense-in-depth pendekatan mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

administrator yang didelegasikan

Di AWS Organizations, layanan yang kompatibel dapat mendaftarkan akun AWS anggota untuk mengelola akun organisasi dan mengelola izin untuk layanan tersebut. Akun ini disebut administrator yang didelegasikan untuk layanan itu. Untuk informasi selengkapnya dan daftar layanan yang kompatibel, lihat Layanan yang berfungsi dengan AWS Organizations AWS Organizations dokumentasi.

deployment

Proses pembuatan aplikasi, fitur baru, atau perbaikan kode tersedia di lingkungan target. Deployment melibatkan penerapan perubahan dalam basis kode dan kemudian membangun dan menjalankan basis kode itu di lingkungan aplikasi.

lingkungan pengembangan

Lihat lingkungan.

kontrol detektif

Kontrol keamanan yang dirancang untuk mendeteksi, mencatat, dan memperingatkan setelah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan kedua, memperingatkan Anda tentang peristiwa keamanan yang melewati kontrol pencegahan di tempat. Untuk informasi selengkapnya, lihat Kontrol Detektif dalam Menerapkan kontrol keamanan pada. AWS

pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik

manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

kembar digital

Representasi virtual dari sistem dunia nyata, seperti bangunan, pabrik, peralatan industri, atau jalur produksi. Kembar digital mendukung pemeliharaan prediktif, pemantauan jarak jauh, dan optimalisasi produksi.

tabel dimensi

Dalam skema bintang, tabel yang lebih kecil yang berisi atribut data tentang data kuantitatif dalam tabel fakta. Atribut tabel dimensi biasanya bidang teks atau angka diskrit yang berperilaku seperti teks. Atribut ini biasanya digunakan untuk pembatasan kueri, pemfilteran, dan pelabelan set hasil.

musibah

Peristiwa yang mencegah beban kerja atau sistem memenuhi tujuan bisnisnya di lokasi utama yang digunakan. Peristiwa ini dapat berupa bencana alam, kegagalan teknis, atau akibat dari tindakan manusia, seperti kesalahan konfigurasi yang tidak disengaja atau serangan malware.

pemulihan bencana (DR)

Strategi dan proses yang Anda gunakan untuk meminimalkan downtime dan kehilangan data yang disebabkan oleh bencana. Untuk informasi selengkapnya, lihat Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML~

Lihat bahasa manipulasi database.

desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani oleh setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

DR

Lihat pemulihan bencana.

deteksi drift

Melacak penyimpangan dari konfigurasi dasar. Misalnya, Anda dapat menggunakan AWS CloudFormation untuk mendeteksi penyimpangan dalam sumber daya sistem, atau Anda dapat menggunakannya AWS Control Tower untuk mendeteksi perubahan di landing zone yang mungkin memengaruhi kepatuhan terhadap persyaratan tata kelola.

DVSM

Lihat pemetaan aliran nilai pengembangan.

E

EDA

Lihat analisis data eksplorasi.

komputasi tepi

Teknologi yang meningkatkan daya komputasi untuk perangkat pintar di tepi jaringan IoT. Jika dibandingkan dengan komputasi awan, komputasi tepi dapat mengurangi latensi komunikasi dan meningkatkan waktu respons.

enkripsi

Proses komputasi yang mengubah data plaintext, yang dapat dibaca manusia, menjadi ciphertext.

kunci enkripsi

String kriptografi dari bit acak yang dihasilkan oleh algoritma enkripsi. Panjang kunci dapat bervariasi, dan setiap kunci dirancang agar tidak dapat diprediksi dan unik.

endianness

Urutan byte disimpan dalam memori komputer. Sistem big-endian menyimpan byte paling signifikan terlebih dahulu. Sistem little-endian menyimpan byte paling tidak signifikan terlebih dahulu.

titik akhir

Lihat titik akhir layanan.

layanan endpoint

Layanan yang dapat Anda host di cloud pribadi virtual (VPC) untuk dibagikan dengan pengguna lain. Anda dapat membuat layanan endpoint dengan AWS PrivateLink dan memberikan izin

kepada prinsipal lain Akun AWS atau ke AWS Identity and Access Management (IAM). Akun atau prinsipal ini dapat terhubung ke layanan endpoint Anda secara pribadi dengan membuat titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat Membuat layanan titik akhir di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).

perencanaan sumber daya perusahaan (ERP)

Sistem yang mengotomatiskan dan mengelola proses bisnis utama (seperti akuntansi, MES, dan manajemen proyek) untuk suatu perusahaan.

enkripsi amplop

Proses mengenkripsi kunci enkripsi dengan kunci enkripsi lain. Untuk informasi selengkapnya, lihat Enkripsi amplop dalam dokumentasi AWS Key Management Service (AWS KMS).

lingkungan

Sebuah contoh dari aplikasi yang sedang berjalan. Berikut ini adalah jenis lingkungan yang umum dalam komputasi awan:

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang dapat diakses oleh pengguna akhir. Dalam pipa CI/CD, lingkungan produksi adalah lingkungan penyebaran terakhir.
- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

epik

Dalam metodologi tangkas, kategori fungsional yang membantu mengatur dan memprioritaskan pekerjaan Anda. Epik memberikan deskripsi tingkat tinggi tentang persyaratan dan tugas implementasi. Misalnya, epos keamanan AWS CAF mencakup manajemen identitas dan akses, kontrol detektif, keamanan infrastruktur, perlindungan data, dan respons insiden. Untuk informasi selengkapnya tentang epos dalam strategi AWS migrasi, lihat panduan implementasi program.

ERP

Lihat perencanaan sumber daya perusahaan.

analisis data eksplorasi (EDA)

Proses menganalisis dataset untuk memahami karakteristik utamanya. Anda mengumpulkan atau mengumpulkan data dan kemudian melakukan penyelidikan awal untuk menemukan pola, mendeteksi anomali, dan memeriksa asumsi. EDA dilakukan dengan menghitung statistik ringkasan dan membuat visualisasi data.

F

tabel fakta

Tabel tengah dalam skema bintang. Ini menyimpan data kuantitatif tentang operasi bisnis. Biasanya, tabel fakta berisi dua jenis kolom: kolom yang berisi ukuran dan yang berisi kunci asing ke tabel dimensi.

gagal cepat

Filosofi yang menggunakan pengujian yang sering dan bertahap untuk mengurangi siklus hidup pengembangan. Ini adalah bagian penting dari pendekatan tangkas.

batas isolasi kesalahan

Dalam AWS Cloud, batas seperti Availability Zone, Wilayah AWS, control plane, atau data plane yang membatasi efek kegagalan dan membantu meningkatkan ketahanan beban kerja. Untuk informasi selengkapnya, lihat Batas Isolasi AWS Kesalahan.

cabang fitur

Lihat cabang.

fitur

Data input yang Anda gunakan untuk membuat prediksi. Misalnya, dalam konteks manufaktur, fitur bisa berupa gambar yang diambil secara berkala dari lini manufaktur.

pentingnya fitur

Seberapa signifikan fitur untuk prediksi model. Ini biasanya dinyatakan sebagai skor numerik yang dapat dihitung melalui berbagai teknik, seperti Shapley Additive Explanations (SHAP) dan gradien terintegrasi. Untuk informasi lebih lanjut, lihat Interpretabilitas model pembelajaran mesin dengan:.AWS

transformasi fitur

Untuk mengoptimalkan data untuk proses ML, termasuk memperkaya data dengan sumber tambahan, menskalakan nilai, atau mengekstrak beberapa set informasi dari satu bidang data. Hal ini memungkinkan model ML untuk mendapatkan keuntungan dari data. Misalnya, jika Anda memecah tanggal “2021-05-27 00:15:37" menjadi “2021", “Mei”, “Kamis”, dan “15", Anda dapat membantu algoritme pembelajaran mempelajari pola bernuansa yang terkait dengan komponen data yang berbeda.

FGAC

Lihat kontrol akses berbutir halus.

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui pengambilan data perubahan untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

G

pemblokiran geografis

Lihat pembatasan geografis.

pembatasan geografis (pemblokiran geografis)

Di Amazon CloudFront, opsi untuk mencegah pengguna di negara tertentu mengakses distribusi konten. Anda dapat menggunakan daftar izinkan atau daftar blokir untuk menentukan negara yang disetujui dan dilarang. Untuk informasi selengkapnya, lihat Membatasi distribusi geografis konten Anda dalam dokumentasi. CloudFront

Alur kerja Gitflow

Pendekatan di mana lingkungan bawah dan atas menggunakan cabang yang berbeda dalam repositori kode sumber. Alur kerja Gitflow dianggap warisan, dan alur kerja berbasis batang adalah pendekatan modern yang disukai.

strategi greenfield

Tidak adanya infrastruktur yang ada di lingkungan baru. Saat mengadopsi strategi greenfield untuk arsitektur sistem, Anda dapat memilih semua teknologi baru tanpa batasan kompatibilitas dengan infrastruktur yang ada, juga dikenal sebagai brownfield. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi (OU). Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

H

HA

Lihat ketersediaan tinggi.

migrasi database heterogen

Memigrasi database sumber Anda ke database target yang menggunakan mesin database yang berbeda (misalnya, Oracle ke Amazon Aurora). Migrasi heterogen biasanya merupakan bagian dari upaya arsitektur ulang, dan mengubah skema dapat menjadi tugas yang kompleks. AWS menyediakan AWS SCT yang membantu dengan konversi skema.

ketersediaan tinggi (HA)

Kemampuan beban kerja untuk beroperasi terus menerus, tanpa intervensi, jika terjadi tantangan atau bencana. Sistem HA dirancang untuk gagal secara otomatis, secara konsisten memberikan kinerja berkualitas tinggi, dan menangani beban dan kegagalan yang berbeda dengan dampak kinerja minimal.

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan

adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

migrasi database homogen

Memigrasi database sumber Anda ke database target yang berbagi mesin database yang sama (misalnya, Microsoft SQL Server ke Amazon RDS for SQL Server). Migrasi homogen biasanya merupakan bagian dari upaya rehosting atau replatforming. Anda dapat menggunakan utilitas database asli untuk memigrasi skema.

data panas

Data yang sering diakses, seperti data real-time atau data translasi terbaru. Data ini biasanya memerlukan tingkat atau kelas penyimpanan berkinerja tinggi untuk memberikan respons kueri yang cepat.

perbaikan terbaru

Perbaikan mendesak untuk masalah kritis dalam lingkungan produksi. Karena urgensinya, perbaikan terbaru biasanya dibuat di luar alur kerja DevOps rilis biasa.

periode hypercare

Segera setelah cutover, periode waktu ketika tim migrasi mengelola dan memantau aplikasi yang dimigrasi di cloud untuk mengatasi masalah apa pun. Biasanya, periode ini panjangnya 1-4 hari. Pada akhir periode hypercare, tim migrasi biasanya mentransfer tanggung jawab untuk aplikasi ke tim operasi cloud.

I

IAc

Lihat infrastruktur sebagai kode.

kebijakan berbasis identitas

Kebijakan yang dilampirkan pada satu atau beberapa prinsip IAM yang mendefinisikan izin mereka dalam lingkungan. AWS Cloud

aplikasi idle

Aplikasi yang memiliki penggunaan CPU dan memori rata-rata antara 5 dan 20 persen selama periode 90 hari. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini atau mempertahankannya di tempat.

IIoT

Lihat Internet of Things industri.

infrastruktur yang tidak dapat diubah

Model yang menyebarkan infrastruktur baru untuk beban kerja produksi alih-alih memperbarui, menambal, atau memodifikasi infrastruktur yang ada. Infrastruktur yang tidak dapat diubah secara inheren lebih konsisten, andal, dan dapat diprediksi daripada infrastruktur yang dapat berubah. Untuk informasi selengkapnya, lihat praktik terbaik Deploy using immutable infrastructure di AWS Well-Architected Framework.

masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

migrasi inkremental

Strategi cutover di mana Anda memigrasikan aplikasi Anda dalam bagian-bagian kecil alih-alih melakukan satu cutover penuh. Misalnya, Anda mungkin hanya memindahkan beberapa layanan mikro atau pengguna ke sistem baru pada awalnya. Setelah Anda memverifikasi bahwa semuanya berfungsi dengan baik, Anda dapat secara bertahap memindahkan layanan mikro atau pengguna tambahan hingga Anda dapat menonaktifkan sistem lama Anda. Strategi ini mengurangi risiko yang terkait dengan migrasi besar.

Industri 4.0

Sebuah istilah yang diperkenalkan oleh Klaus Schwab pada tahun 2016 untuk merujuk pada modernisasi proses manufaktur melalui kemajuan dalam konektivitas, data real-time, otomatisasi, analitik, dan AI/ML.

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAc)

Proses penyediaan dan pengelolaan infrastruktur aplikasi melalui satu set file konfigurasi. IAc dirancang untuk membantu Anda memusatkan manajemen infrastruktur, menstandarisasi sumber daya, dan menskalakan dengan cepat sehingga lingkungan baru dapat diulang, andal, dan konsisten.

Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi selengkapnya, lihat Membangun strategi transformasi digital Internet of Things (IIoT) industri.

inspeksi VPC

Dalam arsitektur AWS multi-akun, VPC terpusat yang mengelola inspeksi lalu lintas jaringan antara VPC (dalam hal yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

Internet of Things (IoT)

Jaringan objek fisik yang terhubung dengan sensor atau prosesor tertanam yang berkomunikasi dengan perangkat dan sistem lain melalui internet atau melalui jaringan komunikasi lokal. Untuk informasi selengkapnya, lihat Apa itu IoT?

interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi selengkapnya, lihat Interpretabilitas model pembelajaran mesin dengan AWS.

IoT

Lihat Internet of Things.

Perpustakaan informasi TI (ITIL)

Serangkaian praktik terbaik untuk memberikan layanan TI dan menyelaraskan layanan ini dengan persyaratan bisnis. ITIL menyediakan dasar untuk ITSM.

Manajemen layanan TI (ITSM)

Kegiatan yang terkait dengan merancang, menerapkan, mengelola, dan mendukung layanan TI untuk suatu organisasi. Untuk informasi tentang mengintegrasikan operasi cloud dengan alat ITSM, lihat panduan integrasi operasi.

ITIL

Lihat perpustakaan informasi TI.

ITSM

Lihat manajemen layanan TI.

L

kontrol akses berbasis label (LBAC)

Implementasi kontrol akses wajib (MAC) di mana pengguna dan data itu sendiri masing-masing secara eksplisit diberi nilai label keamanan. Persimpangan antara label keamanan pengguna dan label keamanan data menentukan baris dan kolom mana yang dapat dilihat oleh pengguna.

landing zone

Landing zone adalah AWS lingkungan multi-akun yang dirancang dengan baik yang dapat diskalakan dan aman. Ini adalah titik awal dari mana organisasi Anda dapat dengan cepat meluncurkan dan menyebarkan beban kerja dan aplikasi dengan percaya diri dalam lingkungan keamanan dan infrastruktur mereka. Untuk informasi selengkapnya tentang zona pendaratan, lihat Menyiapkan lingkungan multi-akun AWS yang aman dan dapat diskalakan.

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat kontrol akses berbasis label.

hak istimewa paling sedikit

Praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM.

angkat dan geser

Lihat 7 Rs.

sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga endianness.

lingkungan yang lebih rendah

Lihat lingkungan.

M

pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat Machine Learning.

cabang utama

Lihat cabang.

malware

Perangkat lunak yang dirancang untuk membahayakan keamanan atau privasi komputer. Malware dapat mengganggu sistem komputer, membocorkan informasi sensitif, atau mendapatkan akses yang tidak sah. Contoh malware termasuk virus, worm, ransomware, Trojan horse, spyware, dan keyloggers.

layanan terkelola

Layanan AWS yang AWS mengoperasikan lapisan infrastruktur, sistem operasi, dan platform, dan Anda mengakses titik akhir untuk menyimpan dan mengambil data. Amazon Simple Storage Service (Amazon S3) dan Amazon DynamoDB adalah contoh layanan terkelola. Ini juga dikenal sebagai layanan abstrak.

sistem eksekusi manufaktur (MES)

Sistem perangkat lunak untuk melacak, memantau, mendokumentasikan, dan mengendalikan proses produksi yang mengubah bahan baku menjadi produk jadi di lantai toko.

PETA

Lihat Program Percepatan Migrasi.

mekanisme

Proses lengkap di mana Anda membuat alat, mendorong adopsi alat, dan kemudian memeriksa hasilnya untuk melakukan penyesuaian. Mekanisme adalah siklus yang memperkuat dan meningkatkan dirinya sendiri saat beroperasi. Untuk informasi lebih lanjut, lihat Membangun mekanisme di AWS Well-Architected Framework.

akun anggota

Semua Akun AWS selain akun manajemen yang merupakan bagian dari organisasi di AWS Organizations. Akun dapat menjadi anggota dari hanya satu organisasi pada suatu waktu.

MES

Lihat sistem eksekusi manufaktur.

Transportasi Telemetri Antrian Pesan (MQTT)

Protokol komunikasi ringan machine-to-machine (M2M), berdasarkan pola terbitkan/berlangganan, untuk perangkat IoT yang dibatasi sumber daya.

layanan mikro

Layanan kecil dan independen yang berkomunikasi melalui API yang terdefinisi dengan baik dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali, dan ketahanan. Untuk informasi selengkapnya, lihat Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server.

arsitektur microservices

Pendekatan untuk membangun aplikasi dengan komponen independen yang menjalankan setiap proses aplikasi sebagai layanan mikro. Layanan mikro ini berkomunikasi melalui antarmuka yang terdefinisi dengan baik dengan menggunakan API ringan. Setiap layanan mikro dalam arsitektur ini dapat diperbarui, digunakan, dan diskalakan untuk memenuhi permintaan fungsi tertentu dari suatu aplikasi. Untuk informasi selengkapnya, lihat Menerapkan layanan mikro di AWS.

Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

migrasi dalam skala

Proses memindahkan sebagian besar portofolio aplikasi ke cloud dalam gelombang, dengan lebih banyak aplikasi bergerak pada tingkat yang lebih cepat di setiap gelombang. Fase

ini menggunakan praktik terbaik dan pelajaran yang dipetik dari fase sebelumnya untuk mengimplementasikan pabrik migrasi tim, alat, dan proses untuk merampingkan migrasi beban kerja melalui otomatisasi dan pengiriman tangkas. Ini adalah fase ketiga dari strategi AWS migrasi.

pabrik migrasi

Tim lintas fungsi yang merampingkan migrasi beban kerja melalui pendekatan otomatis dan gesit. Tim pabrik migrasi biasanya mencakup operasi, analis dan pemilik bisnis, insinyur migrasi, pengembang, dan DevOps profesional yang bekerja di sprint. Antara 20 dan 50 persen portofolio aplikasi perusahaan terdiri dari pola berulang yang dapat dioptimalkan dengan pendekatan pabrik. Untuk informasi selengkapnya, lihat diskusi tentang pabrik migrasi dan panduan Pabrik Migrasi Cloud di kumpulan konten ini.

metadata migrasi

Informasi tentang aplikasi dan server yang diperlukan untuk menyelesaikan migrasi. Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda. Contoh metadata migrasi termasuk subnet target, grup keamanan, dan akun. AWS

pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 AWS dengan Layanan Migrasi Aplikasi.

Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke Cloud. AWS MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). Alat MPA (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

Penilaian Kesiapan Migrasi (MRA)

Proses mendapatkan wawasan tentang status kesiapan cloud organisasi, mengidentifikasi kekuatan dan kelemahan, dan membangun rencana aksi untuk menutup kesenjangan yang diidentifikasi, menggunakan CAF. AWS Untuk informasi selengkapnya, lihat panduan kesiapan migrasi. MRA adalah tahap pertama dari strategi AWS migrasi.

strategi migrasi

Pendekatan yang digunakan untuk memigrasikan beban kerja ke Cloud. AWS Untuk informasi lebih lanjut, lihat entri 7 Rs di glosarium ini dan lihat Memobilisasi organisasi Anda untuk mempercepat migrasi skala besar.

ML

Lihat pembelajaran mesin.

modernisasi

Mengubah aplikasi usang (warisan atau monolitik) dan infrastrukturnya menjadi sistem yang gesit, elastis, dan sangat tersedia di cloud untuk mengurangi biaya, mendapatkan efisiensi, dan memanfaatkan inovasi. Untuk informasi selengkapnya, lihat Strategi untuk memodernisasi aplikasi di. AWS Cloud

penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi selengkapnya, lihat Mengevaluasi kesiapan modernisasi untuk aplikasi di Cloud. AWS

aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat Menguraikan monolit menjadi layanan mikro.

MPA

Lihat Penilaian Portofolio Migrasi.

MQTT

Lihat Transportasi Telemetri Antrian Pesan.

klasifikasi multiclass

Sebuah proses yang membantu menghasilkan prediksi untuk beberapa kelas (memprediksi satu dari lebih dari dua hasil). Misalnya, model ML mungkin bertanya "Apakah produk ini buku, mobil, atau telepon?" atau "Kategori produk mana yang paling menarik bagi pelanggan ini?"

infrastruktur yang bisa berubah

Model yang memperbarui dan memodifikasi infrastruktur yang ada untuk beban kerja produksi. Untuk meningkatkan konsistensi, keandalan, dan prediktabilitas, AWS Well-Architected Framework merekomendasikan penggunaan infrastruktur yang tidak dapat diubah sebagai praktik terbaik.

O

OAC

Lihat kontrol akses asal.

OAI

Lihat identitas akses asal.

OCM

Lihat manajemen perubahan organisasi.

migrasi offline

Metode migrasi di mana beban kerja sumber diturunkan selama proses migrasi. Metode ini melibatkan waktu henti yang diperpanjang dan biasanya digunakan untuk beban kerja kecil dan tidak kritis.

OI

Lihat integrasi operasi.

OLA

Lihat perjanjian tingkat operasional.

migrasi online

Metode migrasi di mana beban kerja sumber disalin ke sistem target tanpa diambil offline. Aplikasi yang terhubung ke beban kerja dapat terus berfungsi selama migrasi. Metode ini melibatkan waktu henti nol hingga minimal dan biasanya digunakan untuk beban kerja produksi yang kritis.

OPC-UA

Lihat Komunikasi Proses Terbuka - Arsitektur Terpadu.

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

Protokol komunikasi machine-to-machine (M2M) untuk otomasi industri. OPC-UA menyediakan standar interoperabilitas dengan enkripsi data, otentikasi, dan skema otorisasi.

perjanjian tingkat operasional (OLA)

Perjanjian yang menjelaskan apa yang dijanjikan kelompok TI fungsional untuk diberikan satu sama lain, untuk mendukung perjanjian tingkat layanan (SLA).

Tinjauan Kesiapan Operasional (ORR)

Daftar pertanyaan dan praktik terbaik terkait yang membantu Anda memahami, mengevaluasi, mencegah, atau mengurangi ruang lingkup insiden dan kemungkinan kegagalan. Untuk informasi lebih lanjut, lihat Ulasan Kesiapan Operasional (ORR) dalam Kerangka Kerja Well-Architected AWS .

teknologi operasional (OT)

Sistem perangkat keras dan perangkat lunak yang bekerja dengan lingkungan fisik untuk mengendalikan operasi industri, peralatan, dan infrastruktur. Di bidang manufaktur, integrasi sistem OT dan teknologi informasi (TI) adalah fokus utama untuk transformasi Industri 4.0.

integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat panduan integrasi operasi.

jejak organisasi

Jejak yang dibuat oleh AWS CloudTrail itu mencatat semua peristiwa untuk semua Akun AWS dalam organisasi di AWS Organizations. Jejak ini dibuat di setiap Akun AWS bagian organisasi dan melacak aktivitas di setiap akun. Untuk informasi selengkapnya, lihat Membuat jejak untuk organisasi dalam CloudTrail dokumentasi.

manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi,

dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat panduan OCM.

kontrol akses asal (OAC)

Di CloudFront, opsi yang disempurnakan untuk membatasi akses untuk mengamankan konten Amazon Simple Storage Service (Amazon S3) Anda. OAC mendukung semua bucket S3 di semua Wilayah AWS, enkripsi sisi server dengan AWS KMS (SSE-KMS), dan dinamis dan permintaan ke bucket S3. `PUT` `DELETE`

identitas akses asal (OAI)

Di CloudFront, opsi untuk membatasi akses untuk mengamankan konten Amazon S3 Anda. Saat Anda menggunakan OAI, CloudFront buat prinsipal yang dapat diautentikasi oleh Amazon S3. Prinsipal yang diautentikasi dapat mengakses konten dalam bucket S3 hanya melalui distribusi tertentu. CloudFront Lihat juga OAC, yang menyediakan kontrol akses yang lebih terperinci dan ditingkatkan.

ORR

Lihat tinjauan kesiapan operasional.

OT

Lihat teknologi operasional.

keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. Arsitektur Referensi AWS Keamanan merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

P

batas izin

Kebijakan manajemen IAM yang dilampirkan pada prinsipal IAM untuk menetapkan izin maksimum yang dapat dimiliki pengguna atau peran. Untuk informasi selengkapnya, lihat Batas izin dalam dokumentasi IAM.

Informasi Identifikasi Pribadi (PII)

Informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar. Contoh PII termasuk nama, alamat, dan informasi kontak.

PII

Lihat informasi yang dapat diidentifikasi secara pribadi.

buku pedoman

Serangkaian langkah yang telah ditentukan sebelumnya yang menangkap pekerjaan yang terkait dengan migrasi, seperti mengirimkan fungsi operasi inti di cloud. Buku pedoman dapat berupa skrip, runbook otomatis, atau ringkasan proses atau langkah-langkah yang diperlukan untuk mengoperasikan lingkungan modern Anda.

PLC

Lihat pengontrol logika yang dapat diprogram.

PLM

Lihat manajemen siklus hidup produk.

kebijakan

Objek yang dapat menentukan izin (lihat kebijakan berbasis identitas), menentukan kondisi akses (lihat kebijakanberbasis sumber daya), atau menentukan izin maksimum untuk semua akun dalam organisasi di (lihat kebijakan kontrol layanan). AWS Organizations

persistensi poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka. Untuk informasi selengkapnya, lihat Mengaktifkan persistensi data di layanan mikro.

penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat Mengevaluasi kesiapan migrasi.

predikat

Kondisi kueri yang mengembalikan `true` atau`false`, biasanya terletak di `WHERE` klausa.

predikat pushdown

Teknik optimasi kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

kontrol preventif

Kontrol keamanan yang dirancang untuk mencegah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan pertama untuk membantu mencegah akses tidak sah atau perubahan yang tidak diinginkan ke jaringan Anda. Untuk informasi selengkapnya, lihat Kontrol pencegahan dalam Menerapkan kontrol keamanan pada. AWS

principal

Entitas AWS yang dapat melakukan tindakan dan mengakses sumber daya. Entitas ini biasanya merupakan pengguna root untuk Akun AWS, peran IAM, atau pengguna. Untuk informasi selengkapnya, lihat Prinsip dalam istilah dan konsep Peran dalam dokumentasi IAM.

Privasi oleh Desain

Pendekatan dalam rekayasa sistem yang memperhitungkan privasi di seluruh proses rekayasa.

zona host pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau beberapa VPC. Untuk informasi selengkapnya, lihat Bekerja dengan zona yang dihosting pribadi di dokumentasi Route 53.

kontrol proaktif

Kontrol keamanan yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat panduan referensi Kontrol dalam AWS Control Tower dokumentasi dan lihat Kontrol proaktif dalam Menerapkan kontrol keamanan pada AWS.

manajemen siklus hidup produk (PLM)

Manajemen data dan proses untuk suatu produk di seluruh siklus hidupnya, mulai dari desain, pengembangan, dan peluncuran, melalui pertumbuhan dan kematangan, hingga penurunan dan penghapusan.

lingkungan produksi

Lihat lingkungan.

pengontrol logika yang dapat diprogram (PLC)

Di bidang manufaktur, komputer yang sangat andal dan mudah beradaptasi yang memantau mesin dan mengotomatiskan proses manufaktur.

pseudonimisasi

Proses penggantian pengenal pribadi dalam kumpulan data dengan nilai placeholder. Pseudonimisasi dapat membantu melindungi privasi pribadi. Data pseudonim masih dianggap sebagai data pribadi.

terbitkan/berlangganan (pub/sub)

Pola yang memungkinkan komunikasi asinkron antara layanan mikro untuk meningkatkan skalabilitas dan daya tanggap. Misalnya, dalam MES berbasis layanan mikro, layanan mikro dapat mempublikasikan pesan peristiwa ke saluran yang dapat berlangganan layanan mikro lainnya. Sistem dapat menambahkan layanan mikro baru tanpa mengubah layanan penerbitan.

Q

rencana kueri

Serangkaian langkah, seperti instruksi, yang digunakan untuk mengakses data dalam sistem database relasional SQL.

regresi rencana kueri

Ketika pengoptimal layanan database memilih rencana yang kurang optimal daripada sebelum perubahan yang diberikan ke lingkungan database. Hal ini dapat disebabkan oleh perubahan statistik, kendala, pengaturan lingkungan, pengikatan parameter kueri, dan pembaruan ke mesin database.

R

Matriks RACI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

ransomware

Perangkat lunak berbahaya yang dirancang untuk memblokir akses ke sistem komputer atau data sampai pembayaran dilakukan.

Matriks RASCI

Lihat bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI).

RCAC

Lihat kontrol akses baris dan kolom.

replika baca

Salinan database yang digunakan untuk tujuan read-only. Anda dapat merutekan kueri ke replika baca untuk mengurangi beban pada database utama Anda.

arsitek ulang

Lihat 7 Rs.

tujuan titik pemulihan (RPO)

Jumlah waktu maksimum yang dapat diterima sejak titik pemulihan data terakhir. Ini menentukan apa yang dianggap sebagai hilangnya data yang dapat diterima antara titik pemulihan terakhir dan gangguan layanan.

tujuan waktu pemulihan (RTO)

Penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan.

refactor

Lihat 7 Rs.

Wilayah

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat Menentukan Wilayah AWS akun yang dapat digunakan.

regresi

Teknik ML yang memprediksi nilai numerik. Misalnya, untuk memecahkan masalah “Berapa harga rumah ini akan dijual?” Model ML dapat menggunakan model regresi linier untuk memprediksi harga jual rumah berdasarkan fakta yang diketahui tentang rumah (misalnya, luas persegi).

rehost

Lihat 7 Rs.

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat 7 Rs.

memplatform ulang

Lihat 7 Rs.

pembelian kembali

Lihat 7 Rs.

ketahanan

Kemampuan aplikasi untuk melawan atau pulih dari gangguan. Ketersediaan tinggi dan pemulihan bencana adalah pertimbangan umum ketika merencanakan ketahanan di. AWS Cloud Untuk informasi lebih lanjut, lihat AWS Cloud Ketahanan.

kebijakan berbasis sumber daya

Kebijakan yang dilampirkan ke sumber daya, seperti bucket Amazon S3, titik akhir, atau kunci enkripsi. Jenis kebijakan ini menentukan prinsipal mana yang diizinkan mengakses, tindakan yang didukung, dan kondisi lain yang harus dipenuhi.

matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI)

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). Jenis dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

kontrol responsif

Kontrol keamanan yang dirancang untuk mendorong remediasi efek samping atau penyimpangan dari garis dasar keamanan Anda. Untuk informasi selengkapnya, lihat Kontrol responsif dalam Menerapkan kontrol keamanan pada AWS.

melestarikan

Lihat 7 Rs.

pensiun

Lihat 7 Rs.

rotasi

Proses memperbarui rahasia secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensil.

kontrol akses baris dan kolom (RCAC)

Penggunaan ekspresi SQL dasar dan fleksibel yang telah menetapkan aturan akses. RCAC terdiri dari izin baris dan topeng kolom.

RPO

Lihat tujuan titik pemulihan.

RTO

Lihat tujuan waktu pemulihan.

buku runbook

Satu set prosedur manual atau otomatis yang diperlukan untuk melakukan tugas tertentu. Ini biasanya dibangun untuk merampingkan operasi berulang atau prosedur dengan tingkat kesalahan yang tinggi.

D

SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke AWS Management Console atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk

semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat Tentang federasi berbasis SAMP 2.0 dalam dokumentasi IAM.

PENIPUAN

Lihat kontrol pengawasan dan akuisisi data.

SCP

Lihat kebijakan kontrol layanan.

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensil pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadata-nya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat Rahasia dalam dokumentasi Secrets Manager.

kontrol keamanan

Pagar pembatas teknis atau administratif yang mencegah, mendeteksi, atau mengurangi kemampuan pelaku ancaman untuk mengeksploitasi kerentanan keamanan. Ada empat jenis kontrol keamanan utama: preventif, detektif, responsif, dan proaktif.

pengerasan keamanan

Proses mengurangi permukaan serangan untuk membuatnya lebih tahan terhadap serangan. Ini dapat mencakup tindakan seperti menghapus sumber daya yang tidak lagi diperlukan, menerapkan praktik keamanan terbaik untuk memberikan hak istimewa paling sedikit, atau menonaktifkan fitur yang tidak perlu dalam file konfigurasi.

sistem informasi keamanan dan manajemen acara (SIEM)

Alat dan layanan yang menggabungkan sistem manajemen informasi keamanan (SIM) dan manajemen acara keamanan (SEM). Sistem SIEM mengumpulkan, memantau, dan menganalisis data dari server, jaringan, perangkat, dan sumber lain untuk mendeteksi ancaman dan pelanggaran keamanan, dan untuk menghasilkan peringatan.

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan detektif atau responsif yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans Amazon EC2, atau memutar kredensil.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCP menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCP sebagai daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam AWS Organizations dokumentasi.

titik akhir layanan

URL titik masuk untuk file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat Layanan AWS titik akhir di Referensi Umum AWS.

perjanjian tingkat layanan (SLA)

Perjanjian yang menjelaskan apa yang dijanjikan tim TI untuk diberikan kepada pelanggan mereka, seperti waktu kerja dan kinerja layanan.

indikator tingkat layanan (SLI)

Pengukuran aspek kinerja layanan, seperti tingkat kesalahan, ketersediaan, atau throughputnya.

tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator tingkat layanan.

model tanggung jawab bersama

Model yang menjelaskan tanggung jawab yang Anda bagikan AWS untuk keamanan dan kepatuhan cloud. AWS bertanggung jawab atas keamanan cloud, sedangkan Anda bertanggung jawab atas keamanan di cloud. Untuk informasi selengkapnya, lihat Model tanggung jawab bersama.

SIEM

Lihat informasi keamanan dan sistem manajemen acara.

titik kegagalan tunggal (SPOF)

Kegagalan dalam satu komponen penting dari aplikasi yang dapat mengganggu sistem.

SLA

Lihat perjanjian tingkat layanan.

SLI

Lihat indikator tingkat layanan.

SLO

Lihat tujuan tingkat layanan.

split-and-seed model

Pola untuk menskalakan dan mempercepat proyek modernisasi. Ketika fitur baru dan rilis produk didefinisikan, tim inti berpisah untuk membuat tim produk baru. Ini membantu meningkatkan kemampuan dan layanan organisasi Anda, meningkatkan produktivitas pengembang, dan mendukung inovasi yang cepat. Untuk informasi lebih lanjut, lihat Pendekatan bertahap untuk memodernisasi aplikasi di. AWS Cloud

SPOF

Lihat satu titik kegagalan.

skema bintang

Struktur organisasi database yang menggunakan satu tabel fakta besar untuk menyimpan data transaksional atau terukur dan menggunakan satu atau lebih tabel dimensi yang lebih kecil untuk menyimpan atribut data. Struktur ini dirancang untuk digunakan dalam gudang data atau untuk tujuan intelijen bisnis.

pola ara pencekik

Pendekatan untuk memodernisasi sistem monolitik dengan menulis ulang secara bertahap dan mengganti fungsionalitas sistem sampai sistem warisan dapat dinonaktifkan. Pola ini menggunakan analogi pohon ara yang tumbuh menjadi pohon yang sudah mapan dan akhirnya mengatasi dan menggantikan inangnya. Pola ini diperkenalkan oleh Martin Fowler sebagai cara untuk mengelola risiko saat menulis ulang sistem monolitik. Untuk contoh cara menerapkan pola ini, lihat Memodernisasi layanan web Microsoft ASP.NET (ASMX) lama secara bertahap menggunakan container dan Amazon API Gateway.

subnet

Rentang alamat IP dalam VPC Anda. Subnet harus berada di Availability Zone tunggal.

kontrol pengawasan dan akuisisi data (SCADA)

Di bidang manufaktur, sistem yang menggunakan perangkat keras dan perangkat lunak untuk memantau aset fisik dan operasi produksi.

enkripsi simetris

Algoritma enkripsi yang menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi data.

pengujian sintetis

Menguji sistem dengan cara yang mensimulasikan interaksi pengguna untuk mendeteksi potensi masalah atau untuk memantau kinerja. Anda dapat menggunakan Amazon CloudWatch Synthetics untuk membuat tes ini.

T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda dapat membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat Menandai AWS sumber daya Anda.

variabel target

Nilai yang Anda coba prediksi dalam ML yang diawasi. Ini juga disebut sebagai variabel hasil. Misalnya, dalam pengaturan manufaktur, variabel target bisa menjadi cacat produk.

daftar tugas

Alat yang digunakan untuk melacak kemajuan melalui runbook. Daftar tugas berisi ikhtisar runbook dan daftar tugas umum yang harus diselesaikan. Untuk setiap tugas umum, itu termasuk perkiraan jumlah waktu yang dibutuhkan, pemilik, dan kemajuan.

lingkungan uji

Lihat lingkungan.

pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan

model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan VPC dan jaringan lokal Anda. Untuk informasi selengkapnya, lihat Apa itu gateway transit dalam AWS Transit Gateway dokumentasi.

alur kerja berbasis batang

Pendekatan di mana pengembang membangun dan menguji fitur secara lokal di cabang fitur dan kemudian menggabungkan perubahan tersebut ke cabang utama. Cabang utama kemudian dibangun untuk pengembangan, praproduksi, dan lingkungan produksi, secara berurutan.

akses tepercaya

Memberikan izin ke layanan yang Anda tentukan untuk melakukan tugas di organisasi Anda di dalam AWS Organizations dan di akunnya atas nama Anda. Layanan tepercaya menciptakan peran terkait layanan di setiap akun, ketika peran itu diperlukan, untuk melakukan tugas manajemen untuk Anda. Untuk informasi selengkapnya, lihat Menggunakan AWS Organizations dengan AWS layanan lain dalam AWS Organizations dokumentasi.

penyetelan

Untuk mengubah aspek proses pelatihan Anda untuk meningkatkan akurasi model ML. Misalnya, Anda dapat melatih model ML dengan membuat set pelabelan, menambahkan label, dan kemudian mengulangi langkah-langkah ini beberapa kali di bawah pengaturan yang berbeda untuk mengoptimalkan model.

tim dua pizza

Sebuah DevOps tim kecil yang bisa Anda beri makan dengan dua pizza. Ukuran tim dua pizza memastikan peluang terbaik untuk berkolaborasi dalam pengembangan perangkat lunak.

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian: ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan

ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data. Untuk informasi lebih lanjut, lihat panduan Mengukur ketidakpastian dalam sistem pembelajaran mendalam.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat lingkungan.

V

menyedot debu

Operasi pemeliharaan database yang melibatkan pembersihan setelah pembaruan tambahan untuk merebut kembali penyimpanan dan meningkatkan kinerja.

kendali versi

Proses dan alat yang melacak perubahan, seperti perubahan kode sumber dalam repositori.

Peering VPC

Koneksi antara dua VPC yang memungkinkan Anda merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat Apa itu peering VPC di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

Cache buffer yang berisi data saat ini dan relevan yang sering diakses. Instance database dapat membaca dari cache buffer, yang lebih cepat daripada membaca dari memori utama atau disk.

data hangat

Data yang jarang diakses. Saat menanyakan jenis data ini, kueri yang cukup lambat biasanya dapat diterima.

fungsi jendela

Fungsi SQL yang melakukan perhitungan pada sekelompok baris yang berhubungan dengan catatan saat ini. Fungsi jendela berguna untuk memproses tugas, seperti menghitung rata-rata bergerak atau mengakses nilai baris berdasarkan posisi relatif dari baris saat ini.

beban kerja

Kumpulan sumber daya dan kode yang memberikan nilai bisnis, seperti aplikasi yang dihadapi pelanggan atau proses backend.

aliran kerja

Grup fungsional dalam proyek migrasi yang bertanggung jawab atas serangkaian tugas tertentu. Setiap alur kerja independen tetapi mendukung alur kerja lain dalam proyek. Misalnya, alur kerja portofolio bertanggung jawab untuk memprioritaskan aplikasi, perencanaan gelombang, dan mengumpulkan metadata migrasi. Alur kerja portofolio mengirimkan aset ini ke alur kerja migrasi, yang kemudian memigrasikan server dan aplikasi.

CACING

Lihat menulis sekali, baca banyak.

WQF

Lihat Kerangka Kualifikasi Beban Kerja AWS.

tulis sekali, baca banyak (WORM)

Model penyimpanan yang menulis data satu kali dan mencegah data dihapus atau dimodifikasi. Pengguna yang berwenang dapat membaca data sebanyak yang diperlukan, tetapi mereka tidak dapat mengubahnya. Infrastruktur penyimpanan data ini dianggap tidak dapat diubah.

Z

eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan zero-day.

kerentanan zero-day

Cacat atau kerentanan yang tak tanggung-tanggung dalam sistem produksi. Aktor ancaman dapat menggunakan jenis kerentanan ini untuk menyerang sistem. Pengembang sering menyadari kerentanan sebagai akibat dari serangan tersebut.

aplikasi zombie

Aplikasi yang memiliki CPU rata-rata dan penggunaan memori di bawah 5 persen. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.