## Best practices for deploying SQL Server on Amazon EC2

# AWS Prescriptive Guidance

# AWS Prescriptive Guidance: Best practices for deploying SQL Server on Amazon EC2

# Table of Contents

# Best practices for deploying Microsoft SQL Server on Amazon EC2

*Abhishek Soni and Sagar Patel, Amazon Web Services (AWS)*

*December 2023* (document history)

The purpose of this guide is to ensure a consistent experience after deploying or migrating Microsoft SQL Server to Amazon Elastic Compute Cloud (Amazon EC2) on the Amazon Web Services (AWS) Cloud. It provides best practices for configuring your database and server, to help optimize your infrastructure, tune performance, and avoid running into unexpected issues after deployment or migration.

This guide is for database architects, systems and database leads, and administrators who are planning to migrate Microsoft SQL Server from their on-premises environment to Amazon EC2, or who want to optimize their new SQL Server deployment on Amazon EC2.

Amazon EC2 provides scalable compute capacity in the AWS Cloud. Using SQL Server on Amazon EC2 is similar to running SQL Server on premises. Amazon EC2 gives you full control over your infrastructure and your database environment. You benefit from the scale, performance, and elasticity of the AWS Cloud, but you're responsible for configuring and tuning all components, including EC2 instances, storage volumes, file systems, networking, and security. This guide provides information to help you optimize your configuration and maximize SQL Server performance on AWS. It discusses server and storage settings and best practices in detail. It also explains how to automate settings where applicable, and discusses configuration changes at the database level.

**Note**

AWS also offers options for moving your on-premises SQL Server database to a managed service like Amazon Relational Database Service (Amazon RDS) for SQL Server. For a discussion of migration options, see Migration strategy for relational databases on the AWS Prescriptive Guidance website.

# Configuring compute and storage settings

Before you migrate or deploy SQL Server on Amazon EC2, you can configure your EC2 instance and storage settings to improve performance and lower your costs. The following sections provide optimization tips and best practices.

**Topics**

## Use an Amazon EBS-optimized instance type

If your SQL Server database handles I/O-intensive workloads, provisioning Amazon Elastic Block Store (Amazon EBS) optimized instances will help improve performance.

An Amazon EBS-optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

## Optimize your disk layout or file distribution

Use one volume for data and log files, another volume for tempdb workloads, and Cold HDD (`sc1`) or Throughput Optimized HDD (`st1`) volumes for backups.

If you run into an I/O-related issue and want to separate workloads for data and log files, consider using different volumes. If your workload requires you to separate specific databases, consider using a dedicated volume for each database.

Typically, tempdb is the target of the highest I/O, so if that workload isn't separated, it might become a bottleneck. This separation also helps isolate tempdb from the data and log files of the user database. You can use comparatively lower-cost storage for backups to optimize your costs.

## Set the NTFS allocation unit size to 64 KB

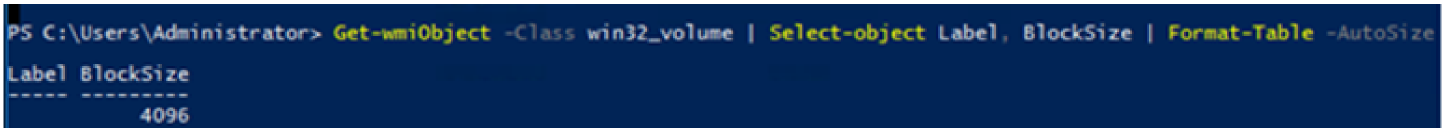
The atomic unit of storage in SQL Server is a *page*, which is 8 KB in size. Eight physically contiguous pages make up an *extent* (which is 64 KB in size). SQL Server uses extents to store data. Therefore, on a SQL Server machine, the NTFS allocation unit size for hosting SQL database files (including tempdb) should be 64 KB.

To check the cluster (NTFS allocation) size of your drives, you can use PowerShell or the command line.

**Using PowerShell:**

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

The following illustration shows example output from PowerShell.



```
PS C:\Users\Administrator> Get-wmiObject -Class win32_volume | Select-object Label, BlockSize | Format-Table -AutoSize
Label BlockSize
----- ---------
           4096
```

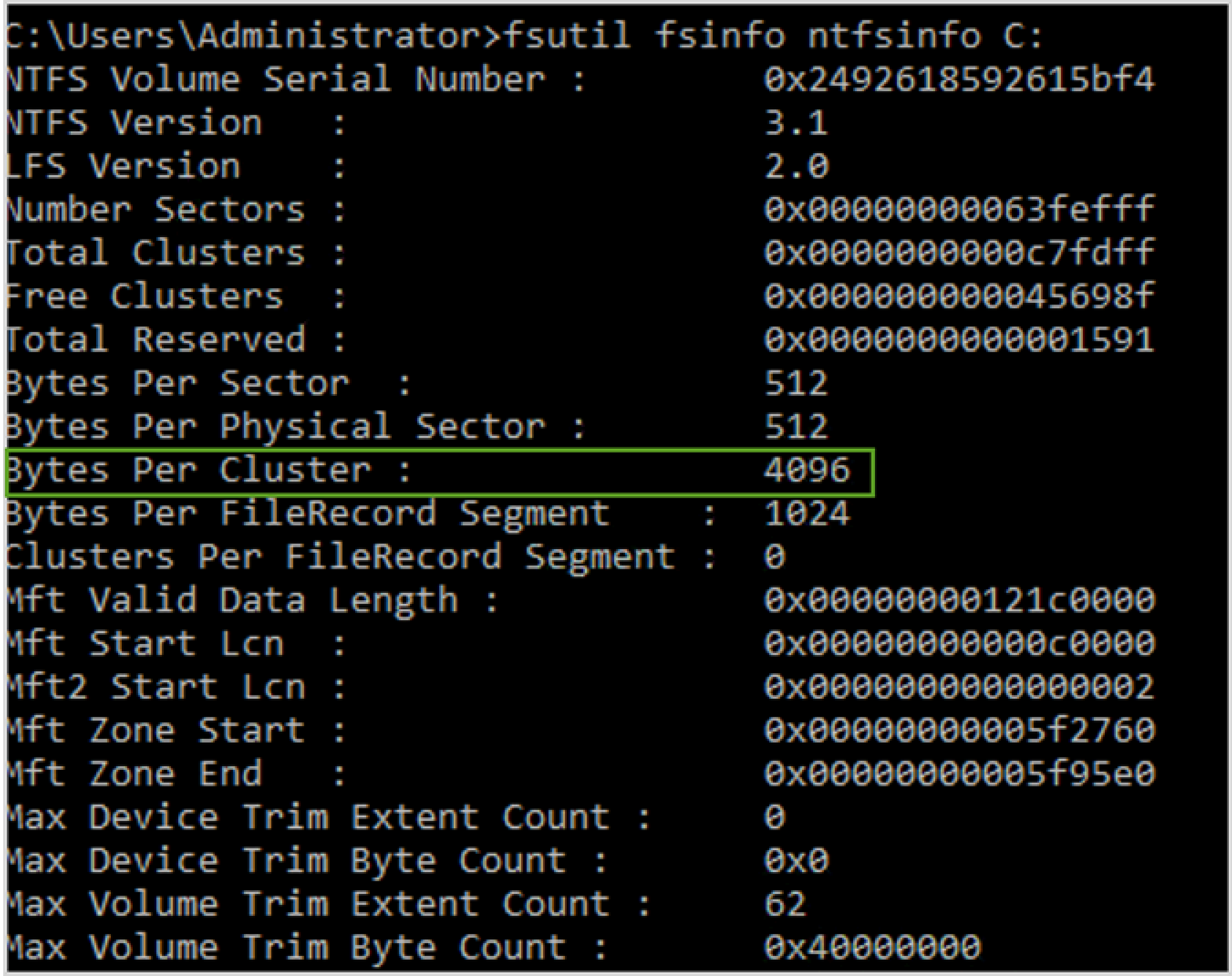
Or use:

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

**Using the command line:**

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

The following illustration shows example output from the command line. The **Bytes Per Cluster** value displays the format size in bytes. The example output shows 4096 bytes. For the drives that host SQL Server database files, this value should be 64 KB.



```
C:\Users\Administrator>fsutil fsinfo ntfsinfo C:
NTFS Volume Serial Number :       0x2492618592615bf4
NTFS Version   :                  3.1
LFS Version    :                  2.0
Number Sectors :                  0x000000000063feff
Total Clusters :                  0x00000000000c7fdff
Free Clusters  :                  0x0000000000045698f
Total Reserved :                  0x0000000000001591
Bytes Per Sector  :               512
Bytes Per Physical Sector :       512
Bytes Per Cluster :               4096
Bytes Per FileRecord Segment    : 1024
Clusters Per FileRecord Segment : 0
Mft Valid Data Length :           0x00000000121c0000
Mft Start Lcn  :                  0x00000000000c0000
Mft2 Start Lcn :                  0x0000000000000002
Mft Zone Start :                  0x000000000005f2760
Mft Zone End   :                  0x000000000005f95e0
Max Device Trim Extent Count :    0
Max Device Trim Byte Count :      0x0
Max Volume Trim Extent Count :    62
Max Volume Trim Byte Count :      0x40000000
```

In some cases, SQL Server performance doesn’t depend on the block size when you use SSD storage on Amazon EC2. For more information, see the blog post Do AWS customers benefit from 64KB block size for SQL Server storage?

## Place tempdb in an instance store

When you use an Amazon EC2 instance store, use the instance store volume for tempdb. An instance store provides temporary (ephemeral) block-level storage for your instance. We recommend that you place tempdb on an instance store volume for two reasons: speed and cost. Tempdb is typically the most heavily used database, so it benefits from the fastest available drive.

Another benefit of placing tempdb in an instance store is cost savings, because you're not charged separately for the I/O against the instance store.

Tempdb is recreated whenever you restart SQL Server, so stopping or terminating an instance won't cause data loss. However, an instance store volume is lost when the virtual machine is started on another host, because the ephemeral disk is attached locally to the machine, so plan carefully.

When you use an instance store volume:

- Initialize the volume before the SQL Server service starts. Otherwise, the SQL Server startup procedure will fail.
- Grant permissions (full control) on the instance store volume explicitly to the SQL Server startup account.

## Moving tempdb to an instance store

**To move tempdb to an instance store volume:**

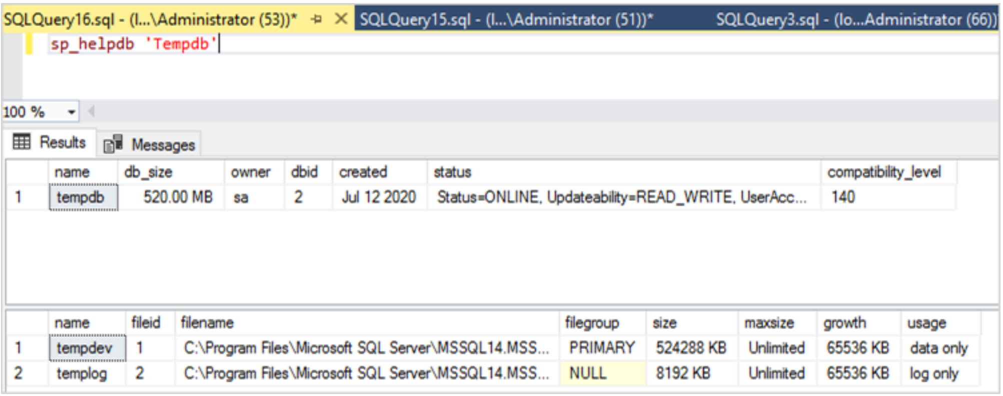
1. From Windows, run `diskmgmt.msc` as an administrator to open the Disk Management system utility.
2. Initialize a new disk.
3. Right-click the disk, and then choose **New Simple Volume**.
4. Complete the prompts, using these settings to format the volume:

   - File system: NTFS
   - Allocation unit size: 64K
   - Volume label: tempdb

   For more information, see the Disk Management documentation on the Microsoft website.
5. Connect to the SQL Server instance, and run the following command to note the logical and physical file name of the tempdb database:

   ```
   $ sp_helpdb 'tempdb'
   ```

   The following screenshot shows the command and its output.


SQLQuery16.sql - (I...\Administrator (53))*
SQLQuery15.sql - (I...\Administrator (51))*
SQLQuery3.sql - (lo...Administrator (66))
100 %
Results
Messages

```
sp_helpdb 'Tempdb'
```

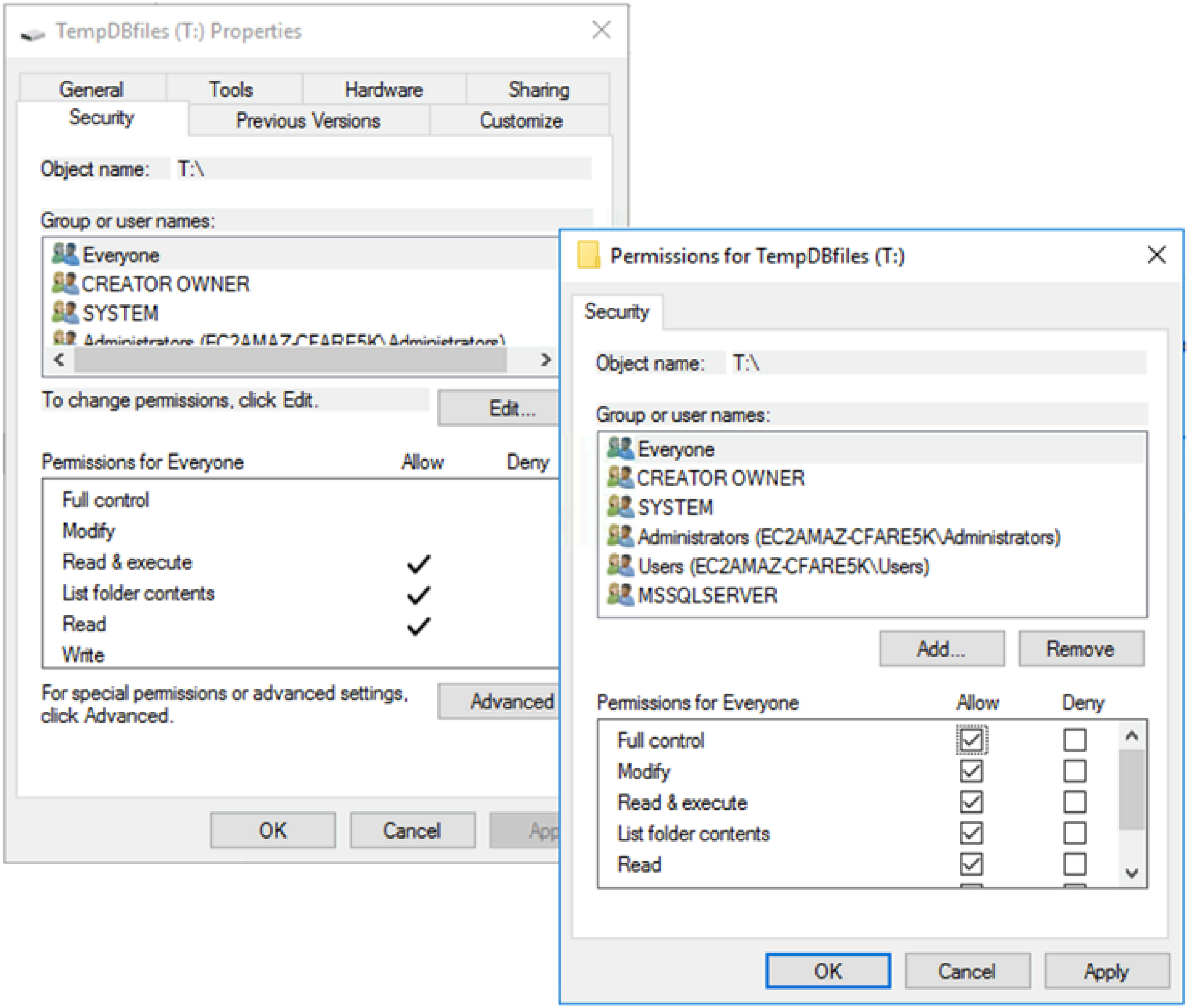
|  | name | db_size | owner | dbid | created | status | compatibility_level |
|---|---|---|---|---|---|---|---|
| 1 | tempdb | 520.00 MB | sa | 2 | Jul 12 2020 | Status=ONLINE, Updateability=READ_WRITE, UserAcc... | 140 |

|  | name | fileid | filename | filegroup | size | maxsize | growth | usage |
|---|---|---|---|---|---|---|---|---|
| 1 | tempdev | 1 | C:\Program Files\Microsoft SQL Server\MSSQL14.MSS... | PRIMARY | 524288 KB | Unlimited | 65536 KB | data only |
| 2 | templog | 2 | C:\Program Files\Microsoft SQL Server\MSSQL14.MSS... | NULL | 8192 KB | Unlimited | 65536 KB | log only |

6. Move the tempdb file to the new location. Remember to set all the tempdb database files to the same initial size. The following sample SQL server script moves the tempdb files to drive T and sets the data files to the same size.

```
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. Grant the SQL Server startup account permissions to the new location of the tempdb database, so it can create the tempdb files, as shown in the following screenshot.



Object name: T:\

Group or user names:

Everyone
CREATOR OWNER
SYSTEM
Administrators (EC2AMAZ-CFARE5K\Administrators)

To change permissions, click Edit.

| Permissions for Everyone | Allow | Deny |
|---|---|---|
| Full control | | |
| Modify | | |
| Read & execute | ✓ | |
| List folder contents | ✓ | |
| Read | ✓ | |
| Write | | |

For special permissions or advanced settings, click Advanced.

Object name: T:\

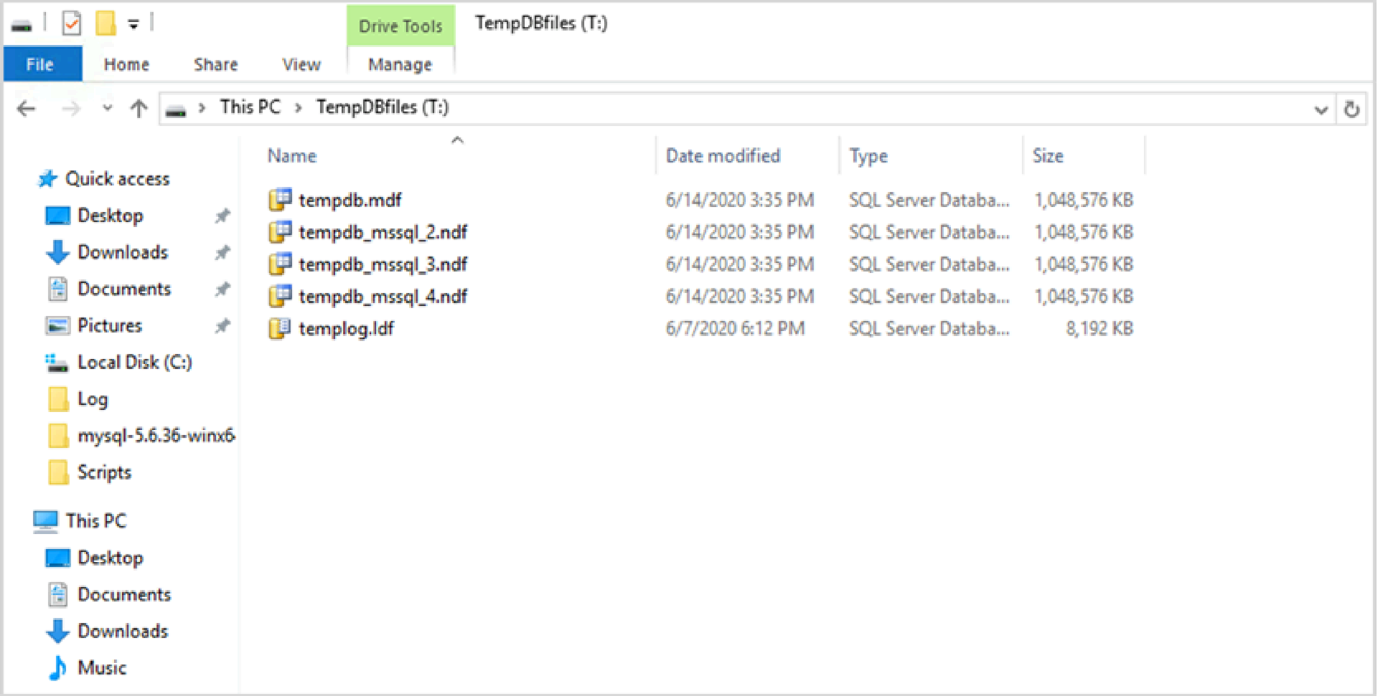
Group or user names:

Everyone
CREATOR OWNER
SYSTEM
Administrators (EC2AMAZ-CFARE5K\Administrators)
Users (EC2AMAZ-CFARE5K\Users)
MSSQLSERVER

| Permissions for Everyone | Allow | Deny |
|---|---|---|
| Full control | ☑ | □ |
| Modify | ☑ | □ |
| Read & execute | ☑ | □ |
| List folder contents | ☑ | □ |
| Read | ☑ | □ |

8. Restart SQL Server to use the new location for tempdb.

   You will see the tempdb files created at the new location, as shown in the following screenshot.



| Name | Date modified | Type | Size |
|---|---|---|---|
| tempdb.mdf | 6/14/2020 3:35 PM | SQL Server Databa... | 1,048,576 KB |
| tempdb_mssql_2.ndf | 6/14/2020 3:35 PM | SQL Server Databa... | 1,048,576 KB |
| tempdb_mssql_3.ndf | 6/14/2020 3:35 PM | SQL Server Databa... | 1,048,576 KB |
| tempdb_mssql_4.ndf | 6/14/2020 3:35 PM | SQL Server Databa... | 1,048,576 KB |
| templog.ldf | 6/7/2020 6:12 PM | SQL Server Databa... | 8,192 KB |

9. Delete the tempdb files from the old location.

To make sure that the instance store volume is initialized before SQL Server starts in case of instance reboots or start/stop, follow the steps in the next section. Otherwise, SQL Server startup will fail because tempdb wasn't initialized.

## Initializing the instance store

**To initialize the data store:**

1. Open the Windows Services Manager (`services.msc`), and set the SQL Server and its dependent services (for example, SQL Server Agent) to start manually. (You will use a script to start it when the instance store volume is ready.)
2. Create a PowerShell script to pass to Amazon EC2 instance as user data. This script does the following:

   - Detects ephemeral storage and creates a tempdb drive for it (drive T in the example).
   - Refreshes the ephemeral disk if the EC2 instance stops and restarts.
   - Grants the SQL Server startup account full control of the newly initialized tempdb volume. The example assumes a default instance, so it uses `NT SERVICE\MSSQLSERVER`. For a named instance, this will typically be `NT SERVICE\MSSQL$`*`<InstanceName>`* by default.

- Saves the script on a local volume (`c:\scripts` in the example) and assigns it a file name (`InstanceStoreMapping.ps1`).
- Creates a scheduled task using Windows Task Scheduler. This task runs the PowerShell script on startup.
- Starts SQL Server and SQL Server Agent after the previous actions.

The following script is from the second lab of the MS-SQL Availability Group Workshop with some changes. Copy the script to the **User** data field when you launch the EC2 instance, and customize it as necessary.

```
<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>
```

## Using the buffer pool extension

If you're planning to use the buffer pool extension, you might also consider placing it on an ephemeral volume. However, we strongly recommend testing it thoroughly before implementation. Avoid using the same volume for the buffer pool extension and tempdb.

**Note**

Although the buffer pool extension can be useful in some cases, it isn't a replacement for RAM. Before you decide to use it, see the details provided on the Microsoft website.

# Avoid CPU core mismatches

Choosing a server that has a higher number of cores than your license covers can cause CPU skew and wasted CPU power. This is because of the mapping between logical and actual cores. When you use SQL Server with a Client Access License (CAL), some schedulers will be `VISIBLE ONLINE` and the rest will be `VISIBLE OFFLINE`. This can lead to performance issues with non-uniform memory access (NUMA) topologies, because of scheduler nodes not being utilized optimally.

For example, if you run SQL Server on an `m5.24xlarge` instance, it will detect two sockets with 24 cores, and 48 logical processors per socket, which results in a total of 96 logical processors. If you have a license for only 48 cores, you would see a message similar to the following in the SQL Server error log:

2020-06-08 12:35:27.37 Server SQL Server detected 2 sockets with 24 cores per socket and 48 logical processors per socket, 96 total logical processors; using 48 logical processors based on SQL Server licensing. This is an informational message; no user action is required.

If you see a difference between the total cores and the number of cores being used by SQL Server, check for CPU usage imbalance or use a server type that has the same number of cores that your license supports.

**CPU skew:** For the instance type in our example (`m5.24xlarge`), SQL Server creates eight NUMA nodes by default. Only four of these nodes (parent node ID 0,1,2,3) have schedulers with the status `VISIBLE ONLINE`. The remaining schedules are all `VISIBLE OFFLINE`. This disparity among schedulers can lead to performance degradation.

To check the scheduler information and status, use:

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

If you want to use a server instance that has a higher number of cores than your SQL Server license supports, consider customizing the number of cores by following the instructions in Specifying CPU options for your instance in the Amazon EC2 documentation.

## Test disk performance

We recommend that you check your disk performance by using a tool like DiskSpd. This tool gives you an estimate of disk speed before you run SQL Server-specific tests. It is very important to evaluate disk performance, because an EBS volume works differently from a traditional SAN in an on-premises environment. The lack of proper performance testing can lead to unexpected slow performance after migration. You can also run custom tests with DiskSpd.

## Enable instant file initialization

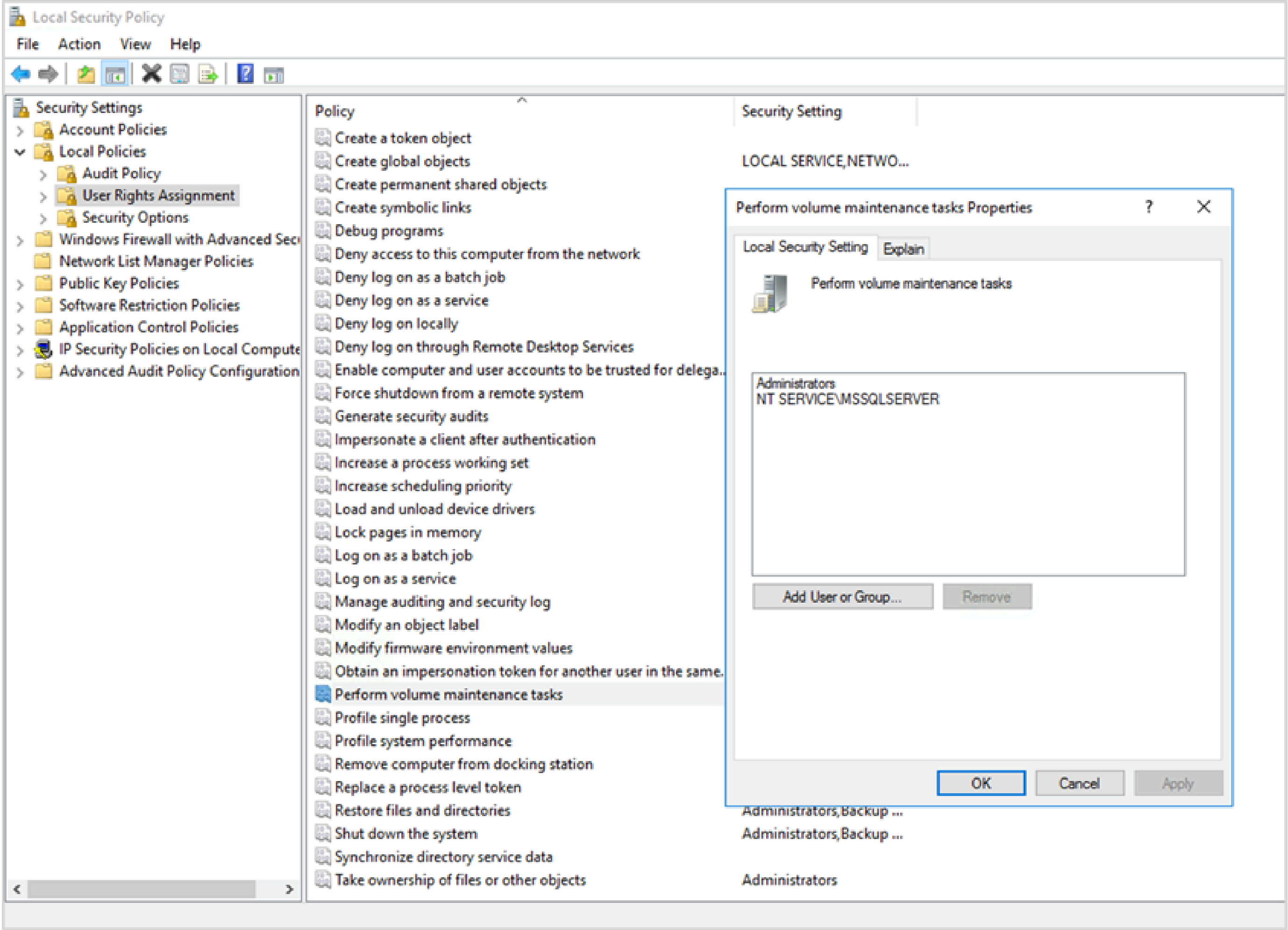
In SQL Server, use the **Perform volume maintenance tasks** setting to enable instant file initialization, unless you're following regulatory restrictions. This option significantly boosts file auto-growth performance.

This setting skips zeroing operations for the data files. That is, data files aren't filled with zero values (`0x0`) when they're initialized, which can take a long time. The current content on the disk is overwritten only when new data is written to the disk.

> **Note**
>
> Log files don't benefit from instant file initialization.

**To enable instant file initialization:**

1. On the **Start** screen, run `secpol.msc` to open the **Local Security Policy** console.
2. Choose **Local Policies**, **User Rights Assignment**, **Perform volume maintenance tasks**, and add the SQL Server service account, as shown in the following screenshot.


Local Security Policy
File
Action
View
Help
Security Settings
Account Policies
Local Policies
Audit Policy
User Rights Assignment
Security Options
Windows Firewall with Advanced Seci
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
IP Security Policies on Local Compute
Advanced Audit Policy Configuration
Perform volume maintenance tasks Properties
?
Local Security Setting
Explain
Perform volume maintenance tasks
Add User or Group...
Remove
OK
Cancel
Apply

| Policy | Security Setting |
|---|---|
| Create a token object | |
| Create global objects | LOCAL SERVICE,NETWO... |
| Create permanent shared objects | |
| Create symbolic links | |
| Debug programs | |
| Deny access to this computer from the network | |
| Deny log on as a batch job | |
| Deny log on as a service | |
| Deny log on locally | |
| Deny log on through Remote Desktop Services | |
| Enable computer and user accounts to be trusted for delega... | |
| Force shutdown from a remote system | |
| Generate security audits | |
| Impersonate a client after authentication | |
| Increase a process working set | |
| Increase scheduling priority | |
| Load and unload device drivers | |
| Lock pages in memory | |
| Log on as a batch job | |
| Log on as a service | |
| Manage auditing and security log | |
| Modify an object label | |
| Modify firmware environment values | |
| Obtain an impersonation token for another user in the same... | |
| Perform volume maintenance tasks | |
| Profile single process | |
| Profile system performance | |
| Remove computer from docking station | |
| Replace a process level token | |
| Restore files and directories | Administrators,Backup ... |
| Shut down the system | Administrators,Backup ... |
| Synchronize directory service data | |
| Take ownership of files or other objects | Administrators |

Administrators
NT SERVICE\MSSQLSERVER

3. Restart the SQL Server instance for changes to take effect.

For more information about instant file initialization, see the SQL Server documentation on the Microsoft website.

**Security note**

When you use instant file initialization, the disk is overwritten only when new data is written to the files, so it’s possible to read deleted content.
While the drive is attached to the instance, the discretionary access control list (DACL) on the file reduces the information disclosure risk, because it allows access only to the SQL Server service account and the local administrator. However, when the file is detached, it can be accessed. If the disclosure of deleted content is a concern, you should disable instant file initialization for the SQL Server instance.

## Lock pages in memory

Enable the **Lock pages in memory** option for the SQL Server startup account to ensure that the operating system doesn’t trim the SQL Server working set.

To check whether this option is enabled, use the following SQL query:

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

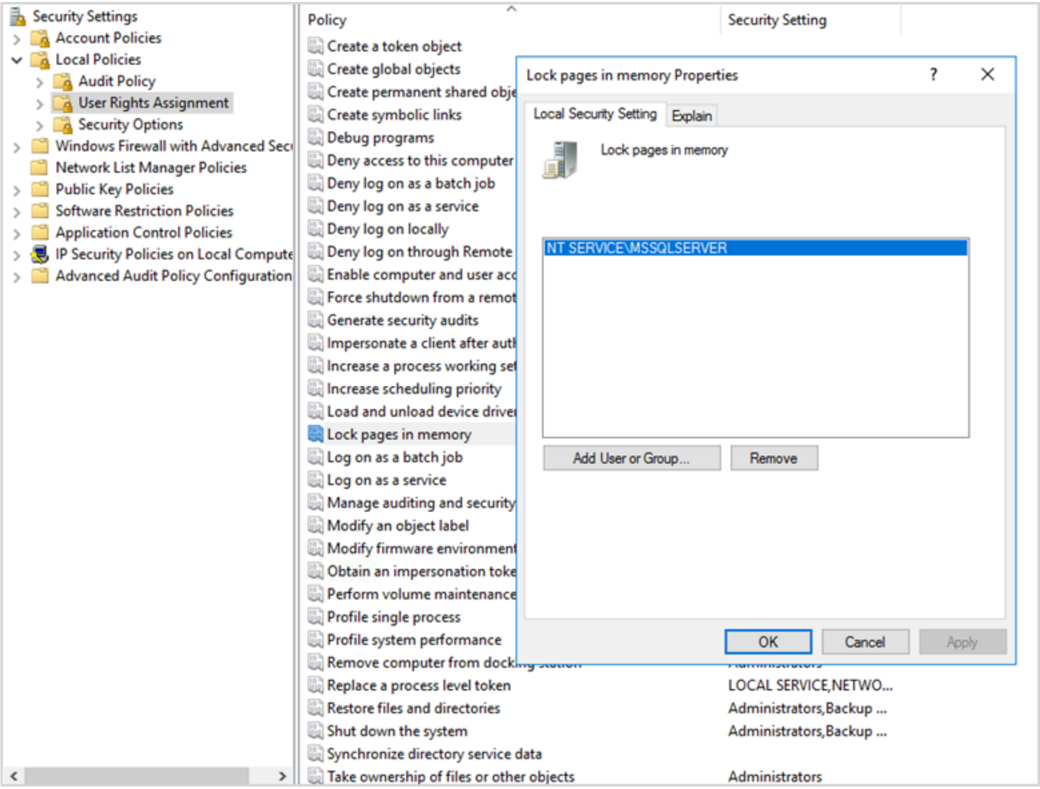
Output:

```
sql_memory_model     sql_memory_model_desc
1                   CONVENTIONAL

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

**To enable the Lock pages in memory option:**

1. On the **Start** screen, run `secpol.msc` to open the **Local Security Policy** console.
2. Choose **Local Policies**, **User Rights Assignment**, **Lock pages in memory**, and add the SQL Server service account, as shown in the following screenshot.


Security Settings
Account Policies
Local Policies
Audit Policy
User Rights Assignment
Security Options
Windows Firewall with Advanced Secu
Network List Manager Policies
Public Key Policies
Software Restriction Policies
Application Control Policies
IP Security Policies on Local Compute
Advanced Audit Policy Configuration
Policy
Security Setting
Create a token object
Create global objects
Create permanent shared obje
Create symbolic links
Debug programs
Deny access to this computer
Deny log on as a batch job
Deny log on as a service
Deny log on locally
Deny log on through Remote
Enable computer and user acc
Force shutdown from a remot
Generate security audits
Impersonate a client after auth
Increase a process working set
Increase scheduling priority
Load and unload device driver
Lock pages in memory
Log on as a batch job
Log on as a service
Manage auditing and security
Modify an object label
Modify firmware environment
Obtain an impersonation toke
Perform volume maintenance
Profile single process
Profile system performance
Remove computer from docking station
Replace a process level token
Restore files and directories
Shut down the system
Synchronize directory service data
Take ownership of files or other objects
LOCAL SERVICE,NETWO...
Administrators,Backup ...
Administrators,Backup ...
Administrators
Lock pages in memory Properties
?
Local Security Setting
Explain
Lock pages in memory
NT SERVICE\MSSQLSERVER
Add User or Group...
Remove
OK
Cancel
Apply

3. Restart the SQL Server instance for changes to take effect.

4. Use the following SQL query to confirm that the **Lock pages in memory** option is enabled:

```
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.
```

For more information about the SQL Server memory model, see `sql_memory_model` and `sql_memory_model_desc` in the sys.dm_os_sys_info documentation on the Microsoft website.

# Disable TCP offloading and RSS settings

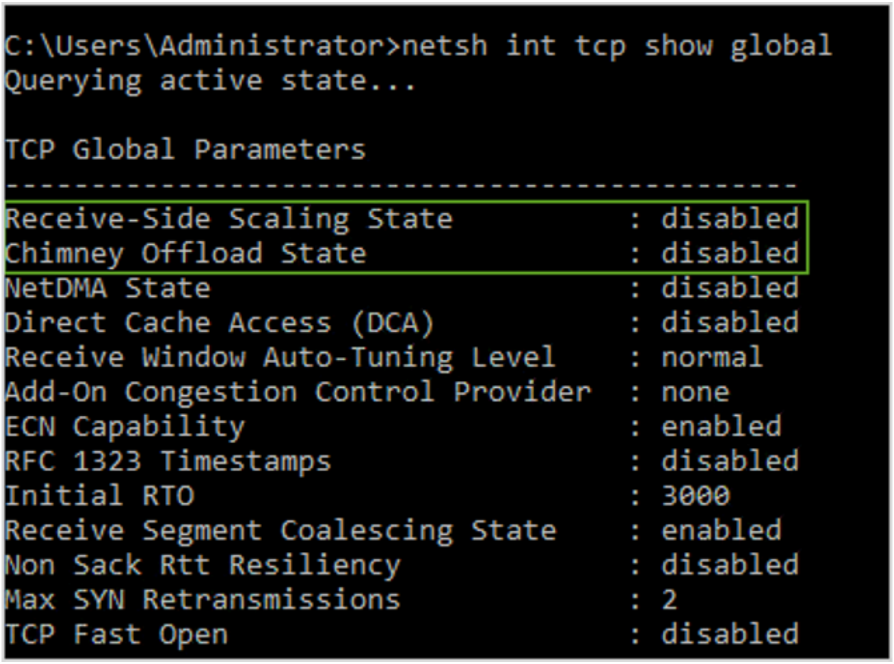
If you observe random connectivity issues such as transport-level errors or packet transmission errors when you run SQL workloads, you might want to disable TCP offloading and RSS settings.

- TCP offloading (**TCP Chimney Offload** feature) offloads the processing of TCP/IP packets from the processor to the network adapter, to free the CPU for other tasks.
- Receive-side scaling (RSS) helps distributes the processing of incoming network traffic on multiprocessor systems. It load balances the network processing efficiently among the CPUs.

To check your current settings, at the command prompt, run the **netsh** command:

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

Here is sample output from the command. In this example, **Receive-Side Scaling State** and **Chimney Offload State** are both disabled.



```
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
```

To get task offload information about a specific connection, at the command prompt, run:

```
netstat –t
```

and check the value of the offload state column.

To disable TCP offloading and RSS for Windows Server 2008 and 2012, run these commands at the command prompt:

```
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
```

To learn more about these settings, see:

- TCP Chimney Offload, Receive Side Scaling, and Network Direct Memory Access features and Introduction to Receive Side Scaling on the Microsoft website
- TCP Offloading in the Amazon EC2 documentation
- Troubleshooting PV Drivers in the Amazon EC2 documentation

> ⚠ **Important**
>
> Do not use **IPsec Task Offload** or **TCP Chimney Offload**. According to the Microsoft documentation, these offload features are deprecated in Windows Server 2016 and might not be supported in future versions. Using these features might adversely affect performance.

## Determine your IOPS and throughput requirements

Use Windows Performance Monitor to get information about IOPS and throughput.

To open Windows Performance Monitor, run **perfmon** at the command prompt. IOPS and throughput data is provided by the following performance counters:

- Disk reads/sec + disk writes/sec = IOPS
- Disk read bytes/sec + disk write bytes/sec = throughput

We recommend that you get the IOPS and throughput data for peak usage time and also over a typical workload cycle, to get a good estimate of your requirements. Make sure that the instance type you choose for SQL Server supports these I/O requirements.

It is important to get this estimate right. Otherwise, you might over-provision your resources, which might result in underutilized resources, or under-provision your resources, which might result in severe performance issues.

## Use striping to bypass IOPS and throughput limitations

When your SQL Server application requires more than the maximum IOPS and throughput available on an EBS volume, consider striping your EBS volume to overcome these limitations.

Striping volumes (RAID) helps you meet your IOPS and throughput requirements, and is limited by the maximum IOPS and bandwidth supported by a particular instance. For more information about striping options, see RAID configuration in the Amazon EC2 documentation. You can also use Storage Spaces on a standalone server. For more information, see the Microsoft documentation.

## Exclude SQL Server files from antivirus software

When you configure your antivirus software settings, make sure that you exclude your SQL Server files and directories from virus scanning. For details and a list of files and directories to exclude, see How to choose antivirus software to run on computers that are running SQL Server on the Microsoft website.

If you don't exclude these SQL Server files, they could be corrupted or quarantined by antivirus software when SQL Server needs to use them. Not excluding these files can also cause performance problems.

# Configuring SQL Server

This section provides best practices for configuring your SQL Server databases to fine-tune performance, avoid common pitfalls, and meet your security and availability requirements. You can implement these changes before or after you migrate your databases to Amazon EC2. The following sections provide configuration tips and best practices.

**Topics**

## Configure tempdb to reduce contention

We recommend that you configure tempdb with multiple data files of equal size and with equal growth factor.

On a busy database server that uses tempdb a lot, you might notice severe blocking when the server experiences a heavy load. You might notice that tasks are waiting for wait resources that point to pages in tempdb. These pages might be Page Free Space (PFS) and Shared Global Allocation Map (SGM) pages that have the format 2:*x*:*x* (for example, 2:1:1 or 2:1:2).

To improve the concurrency of tempdb, you can increase the number of data files in tempdb to maximize disk bandwidth and reduce contention in allocation structures. Here are some guidelines:

- If the number of logical processors is equal to, or less than, 8: Use the same number of data files and logical processors.
- If the number of logical processors is higher than 8: Use 8 data files.

If contention persists, increase the number of data files in multiples of 4 until the contention is remediated, up to the number of logical processors on the server. This will help avoid SGAM contention in tempdb. If you're using SQL Server 2014 or an earlier version, you also need to enable trace flag 1118. This flag forces page allocations on uniform extents instead of mixed extents, which minimizes scans on the SGAM page and reduces contention.

Starting with SQL Server 2016 (13.x), this behavior is controlled by the `AUTOGROW_SINGLE_FILE` and `AUTOGROW_ALL_FILES` options of `ALTER DATABASE`. For example:

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

For more information about setting these options, see the Microsoft SQL Server documentation.

## Set MAXDOP for best performance

The maximum degree of parallelism (MAXDOP) is a server configuration option for running SQL Server on multiple CPUs. It controls the number of processors used to run a single statement in parallel plan execution. The default value is 0, which enables SQL Server to use all available processors. This can affect performance, and isn't optimal for most use cases.

Use the following guidelines when you configure the MAXDOP value for SQL Server.

| NUMA nodes | Logical processors | MAXDOP value |
|---|---|---|
| Single | ≤ 8 | 4, 2, or number of cores (for one or two cores) |
| Single | > 8 | 8, 4, or 2 |
| Multiple | ≤ 16 | 8, 4, or 2 |
| Multiple | > 16 | 16, 8, 4, or 2 |

**Note**

Setting MAXDOP to 2, 4, or 8 generally provides the best results in most use cases. We recommend that you test your workload and monitor for any parallelism-related wait types such as CXPACKET.

You can use the following query to gather the current NUMA configuration for SQL Server 2016 and later versions:

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

where:

- `cpu_count` refers to the number of logical CPUs in the system.
- `hyperthread_ratio` is the ratio of the number of cores that are exposed by one physical processor.
- `softnuma_configuration` is `0`, 1, or 2:
  - `0 (OFF)`: default
  - `1 (automated)`: soft-NUMA
  - `2 (manual)`: soft-NUMA
- `softnuma_configuration_desc` is `OFF`, `ON`, or `MANUAL`:
  - `OFF` indicates that the soft-NUMA feature is off.
  - `ON` indicates that SQL Server automatically decides the NUMA node sizes.
  - `MANUAL` indicates that soft-NUMA is manually configured.
- `socket_count` is the number of processor sockets.
- `numa_node_count` is the number of NUMA nodes available in the system.

To check the current MAXDOP value, use:

```
$ sp_configure 'max_degree_of_parallelism'
```

For more information about MAXDOP, see the Microsoft SQL Server documentation.

# Change the cost threshold of parallelism

The cost threshold of parallelism determines which queries are candidates for parallel execution. This property’s default value is 5, which means that the optimizer switches to a parallel plan if the cost of a serial plan is more than 5 (which refers to an abstracted unit of cost, not estimated time). We recommend that you set this property to a higher number.

The default value was appropriate back when processors had high price tags, processing power was low, and query processing was slower than it is now. Processors today are much faster. As a result, comparatively smaller queries (for example, given a cost threshold of 32) won’t benefit a lot from parallel execution, especially given the overhead associated with coordinating parallel execution.

In most cases, a cost threshold of parallelism setting of 50 is a good starting point. Here's an example of how to configure the cost threshold of parallelism:

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

# Optimize for ad hoc workloads

Enable the **optimize for ad hoc workloads** option to improve the efficiency of the plan cache for workloads that contain many single-use, ad hoc batches. When you initially run an ad hoc query, the database engine caches a compiled plan stub instead of the complete execution plan, thus saving space in the plan cache. If you run the ad hoc batch again, the database engine recognizes that the batch has been run before and replaces the compiled plan stub with the full, compiled plan in the plan cache.

To check whether this option is enabled, use the query:

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

For more information about optimizing for ad hoc workloads, see the Microsoft SQL Server documentation.

## Use trace flags to improve performance

Consider using SQL Server trace flags that are applicable to your environment to enhance performance. For example:

- **4199:** Enables query optimizer (QO) changes that are released in SQL Server Cumulative Updates (CUs) and Service Packs (SPs).
- **8048:** Converts NUMA-partitioned memory objects to CPU-partitioned memory objects.
- **9024:** Converts a global log pool memory object to a NUMA-partitioned memory object.

The following examples illustrate how to turn trace flags on and off for SQL Server on Amazon EC2. If you experience any issues when you enable tracing, make sure that you have the appropriate permissions for the account.

To turn trace flag 4199 on, run:

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

To check the status of the trace flag, run:

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

To turn trace flag 4199 off, run:

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

For a complete list of trace flags, see the Microsoft SQL Server documentation.

## Install the latest patches

Starting with SQL Server 2017, Microsoft has discontinued the release of Service Packs (SPs). It releases Cumulative Updates (CUs) and critical updates (GDRs) only.

SPs include crucial fixes for SQL Server, so make sure that the latest SP has been installed. Also, if possible, install the latest CU package.

For information about the latest SQL Server updates, see Latest updates for Microsoft SQL Server on the Microsoft website.

## Cap max server memory to avoid memory pressure

For performance reasons, SQL Server doesn’t release memory that it has already allocated. When SQL Server is started, it slowly takes the memory specified under the **min_server_memory** option, and then continues to grow until it reaches the value specified in the **max_server_memory** option. (For more information about these settings, see Server memory configuration options in the SQL Server documentation.)

SQL Server memory has two components: the buffer pool and the non-buffer pool (also called *memory to leave* or MTL). The value of the **max_server_memory** option determines the size of the SQL Server buffer pool, which consists of the buffer cache, procedure cache, plan cache, buff structures, and other caches.

Starting with SQL Server 2012, **min_server_memory** and **max_server_memory** account for all memory allocations for all caches, including `SQLGENERAL`, `SQLBUFFERPOOL`, `SQLQUERYCOMPILE`, `SQLQUERYPLAN`, `SQLQUERYEXEC`, `SQLOPTIMIZER`, and `SQLCLR`. For a complete list of memory clerks under max_server_memory, see sys.dm_os_memory_clerks in the Microsoft SQL Server documentation.

To check the current **max_server_memory** value, use the command:

```
$ sp_configure 'max_server_memory'
```

We recommend that you cap **max_server_memory** at a value that doesn’t cause systemwide memory pressure. There’s no universal formula that applies to all environments, but we have provided some guidelines in this section. **max_server_memory** is a dynamic option, so it can be changed at run time.

As a starting point, you can determine **max_server_memory** as follows:

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

where:

- Memory for the operating system is 1-4 GB.
- MTL (memory to leave) includes the stack size, which is 2 MB on 64-bit machines per worker thread and can be calculated as follows: `MTL = stack_size * max_worker_threads`

Alternatively, you can use:

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

where the memory basis amount of RAM is determined as follows:

- If RAM on the server is between 4 GB and 16 GB, leave 1 GB per 4 GB of RAM. For example, for a server with 16 GB, leave 4 GB.
- If RAM on the server is over 16 GB, leave 1 GB per 4 GB of RAM up to 16 GB, and 1 GB per 8 GB of RAM above 16 GB.

For example, if a server has 256 GB of RAM, the calculation will be:

- 1 GB for the OS
- Up to 16 GB RAM: 16/4 = 4 GB
- Remaining RAM above 16 GB: (256-16)/8 = 30
- Total RAM to leave: 1 + 4 + 30 = 35 GB
- max_server_memory: 256 - 35 = 221 GB

After initial configuration, monitor the memory you can free over a typical workload duration to determine if you need to increase or decrease the memory allocated to SQL Server.

> **Note**
>
> Windows signals the low memory resource notification at 96 MB, so you want a buffer, but you can set **Available Mbytes** to be above 1 GB on larger servers with 256 GB or higher RAM.

For additional information, see the Memory Management Architecture Guide in the Microsoft SQL Server documentation.

## Use the highest database compatibility level

Check to make sure that you're using the current database compatibility level to take advantage of the latest improvements in SQL Server. This is important to check because when you restore a database from a lower version to a higher version, it will keep the compatibility level of the lower version. Some of the latest database enhancements are effective only when you set database compatibility to the most recent level available for the engine version you installed.

To check the current database compatibility use:

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

For more information about database compatibility levels, see the Microsoft SQL Server documentation.

## Control the number of VLFs

Pre-allocate the maximum size of data and log files. For better performance, control the number of virtual log files (VLFs) by pre-allocating the space and correcting the automatic growth (autogrow) settings for log files.

Typically, an autogrow factor of 8 GB works well in most production environments. Consider growing the transaction log files in 8-GB chunks. A higher number of VLFs can extend the backup and recovery time for the database, and can cause performance issues with any operation (for example, replication) that requires going through the log files.

For more information about the VLF creation and growth algorithm, see the SQLskills blog.

## Check database autogrowth settings

Any transaction that needs the data or the log file to grow includes the time taken by the file growth operation. The file grows by the increment size defined by the **FILEGROWTH** option. You can look for file growth events in SQL Server profiler traces. If file growth takes a long time, you might see wait types like `ASYNC_IO_COMPLETION`, which occurs when data processing is very slow. Such wait types not only affect performance but might also result in transaction timeouts. If that transaction holds locks on resources sought by other transactions, the timeout would lead to severe server blocking issues.

For this reason, we recommend that you configure autogrowth settings very carefully. Also keep in mind that:

- File growth is one of the costliest operations in SQL Server.
- Frequent autogrowth in small chunks can lead to disk fragmentation.
- Frequent autogrowth in log files results in a large number of virtual log files (VLFs) and affects performance, as discussed in the previous section.

All these reasons could lead to slow database startup and increased backup and recovery time.

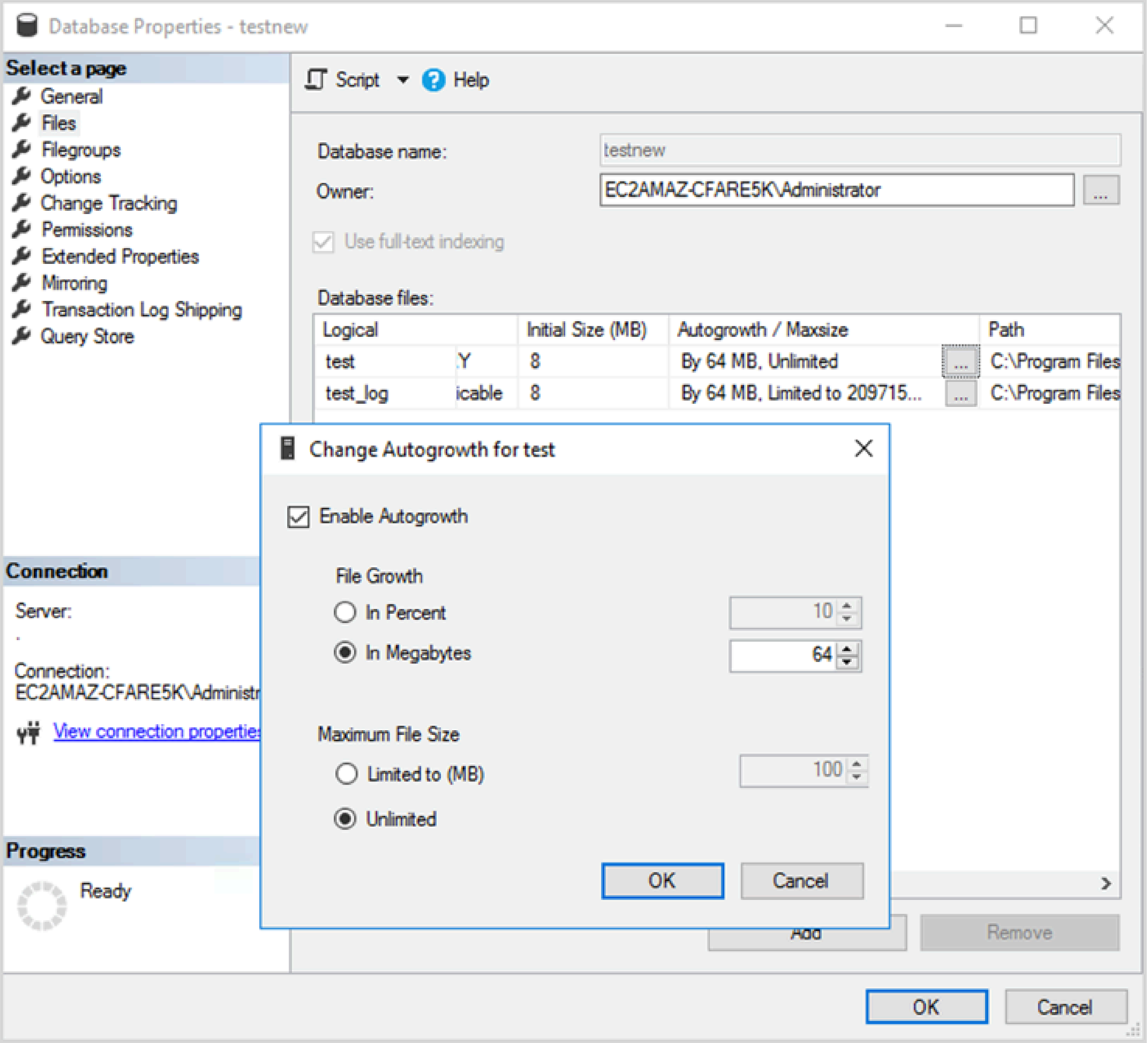
Ideally, you should pre-grow files proactively, based on regular monitoring. Choose carefully between setting autogrowth as a percentage or as a static value (in MB). Typically, setting autogrowth to one eighth of the file size is a good starting point, but this might not be the right choice. (For example, this percentage would be too high if your data file is several TBs in size.)

In most cases, an autogrowth value of 1024 MB works well for data files in most large databases. For log files, 512 MB is a good starting point. For contingency measures, we strongly recommend that you set the autogrowth value, but grow the files manually for a few months based on past trends.

**Note**

Setting autogrowth should be a contingency measure, so you should set it after you pre-allocate storage to a file.

You can change autogrowth settings by using SQL Server Management Studio (SSMS) or Transact-SQL. The following screen illustration shows autogrowth settings in SSMS.



Database name: testnew

Owner: EC2AMAZ-CFARE5K\Administrator

☑ Use full-text indexing

Database files:

| Logical | | Initial Size (MB) | Autogrowth / Maxsize | | Path |
|---|---|---|---|---|---|
| test | Y | 8 | By 64 MB, Unlimited | ... | C:\Program Files |
| test_log | icable | 8 | By 64 MB, Limited to 209715... | ... | C:\Program Files |

Server:
.

Connection:
EC2AMAZ-CFARE5K\Administr

Change Autogrowth for test

☑ Enable Autogrowth

File Growth

○ In Percent 10

● In Megabytes 64

Maximum File Size

○ Limited to (MB) 100

● Unlimited

When you use the **FILEGROWTH** option for data and log files, choose carefully between setting it as a percentage or as a static value (in MB). Setting a percentage results in ever-increasing file growth, so you might prefer to use a static size for better control over the growth ratio.

- In versions before SQL Server 2022 (16.x), transaction logs cannot use instant file initialization, so extended log growth times are especially critical.
- Starting with SQL Server 2022 (16.x, all editions), instant file initialization can benefit transaction log growth events up to 64 MB. The default autogrowth size increment for new

databases is 64 MB. Transaction log file autogrowth events that are larger than 64 MB cannot benefit from instant file initialization.

# Configuring Always On availability groups

If you're using native client libraries for SQL Server version 2012 and later, and .NET Framework 4.5 libraries, you can use the **MultiSubnetFailover** parameter to change the connection behavior. We recommend that you set this parameter to TRUE. This will enable faster failover with Always On availability groups.

> **Note**
>
> If you have legacy applications that cannot use the **MultiSubnetFailover** parameter, you can place a Network Load Balancer in front of your SQL Server instances. The balancer uses a health check that determines which SQL Server database is active and sends traffic to the instance that currently hosts that database. The load balancer spans one or multiple Availability Zones. You can use a dedicated port such as 59999 for the health check, and then modify the cluster group parameter to respond to that port. This enables you to reduce SQL Server failover time to approximately one minute without using the **MultiSubnetFailover** parameter. For detailed instructions, see the blog post Reduce failover times for SQL Server on Amazon EC2 instance using Network Load Balancer.

Two settings affect how the availability group listener is registered with DNS: **RegisterAllProvidersIP** and **HostRecordTTL**.

## Set RegisterAllProvidersIP to true when using Always On availability groups

We recommend that you set **RegisterAllProvidersIP** to 1 (**true**). When the availability group listener is created with **RegisterAllProvidersIP** set to 1, all IP addresses for that listener are registered in DNS. When **RegisterAllProvidersIP** is set to `0` (**false**), only one active IP is registered.

In case of failover, when the primary replica moves from one subnet to the other, the old IP address is unregistered, and the new IP address is registered. The DNS is updated with the new IP when the availability group listener comes online. However, client systems won't resolve the listener name to the new IP address until the currently cached entry expires.

## Set HostRecordTTL to 60 or less when using Always On availability groups

The **HostRecordTTL** setting controls the Time to Live (TTL) for cached DNS entries. The default value is 1200 seconds. We recommend that you change **HostRecordTTL** to a much lower setting (60 seconds or less). This causes the cached value to expire sooner, so in case of failover, client systems can resolve the new IP more quickly.

## Disable automatic failback for the Always On cluster group

Verify that automatic failback is disabled for the Always On availability groups in the Windows Cluster Manager.

# Configuring backups

As discussed in the Optimize your disk layout or file distribution section, we recommend that you send your native SQL Server backups to a separate drive. Also consider taking a scheduled snapshot of the EBS volume where the backup files reside.

# Improving database optimization

This section provides best practices for improving performance when working with the SQL Server query optimizer. It discusses how rebuilding indexes and updating table statistics on a regular basis can help optimize execution plans. The following sections provide configuration tips and best practices.

**Topics**

- Rebuild indexes
- Update statistics

## Rebuild indexes

For the query optimizer to generate the best possible query plans and use the right indexes, the indexes shouldn't be fragmented. Indexes become fragmented over time based on the update, insert, or delete rate. Make sure that tables are re-indexed on a regular basis. The rebuild frequency depends on the rate at which the database handles data manipulation language (DML) operations.

A good starting point would be to rebuild indexes that are fragmented more than 30%, and reorganize indexes that are fragmented less than 30%. The 30% value works in a majority of use cases, but if you still see poor query plans due to unused indexes, you might need to revisit this percentage.

Use a query like the following to check for fragmentation:

```
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
```

We recommend that you create a maintenance job to rebuild the indexes on a regular basis.

## Update statistics

As with fragmented indexes, if the optimizer doesn't have up-to-date information about the distribution of key values (statistics) of table columns, it cannot generate optimal execution plans.

We recommend that you update the statistics for all tables on a regular basis. The frequency of updates depends on the rate at which the database handles DML operations, but is typically run twice a week during off peak hours. However, avoid updating statistics on the days that you’re rebuilding indexes. For more information on updating statistics, see the Microsoft SQL Server documentation.

For database optimization, we recommend using an index and statistics maintenance script. For an example, see the SQL Server index and statistics maintenance script provided on the SQL Server Maintenance Solution website.

# Optimizing SQL Server deployments on Amazon EC2 with AWS Launch Wizard

AWS Launch Wizard is the primary method for SQL Server single instance and high availability (HA) deployments on Amazon EC2. Launch Wizard deployments are based on the AWS Well-Architected Framework, and are optimized for security, reliability, performance efficiency, and cost savings.

Launch Wizard simplifies your SQL Server deployments and also makes it easier to configure SQL Server. Its features include:

- **Automated AWS resource selection –** Launch Wizard can recommend the optimal instance type based on your virtual CPU (vCPU), memory, and networking requirements. It can also recommend the volume type based on the storage drive and throughput.
- **One-click monitoring –** Launch Wizard integrates with Amazon CloudWatch Application Insights to set up monitoring for SQL Server HA deployments on AWS. When you select this option, Application Insights automatically sets up relevant metrics, logs, and alarms on CloudWatch, and starts monitoring newly deployed workloads.
- **Application resource groups for easy discoverability –** Launch Wizard creates a resource group for all AWS resources that were created for your SQL Server application. You can manage, patch, and maintain your SQL Server applications from the AWS Systems Manager console.

Launch Wizard provides you with reusable AWS CloudFormation code templates. These templates can serve as a baseline for your subsequent application deployments. To learn more, see the AWS Launch Wizard overview and user guide.

# Next steps

This guide covered some of the best practices for configuring and running Microsoft SQL Server workloads on Amazon EC2. Following these guidelines in the planning and implementation phases of the migration process will help you establish a stable server in the production environment.

For more information about these configuration tasks, see the links provided in each section and visit the webpages listed in the Additional resources section.

# Additional resources

### Related strategies, guides, and patterns

- Migration strategy for relational databases
- *Patterns* section of the AWS Prescriptive Guidance website (you can use the filters on that page to view patterns by migration strategy, such as re-architect, rehost, relocate, or replatform)

### AWS resources

- AWS documentation
- AWS general reference
- AWS glossary

### AWS services

- Amazon EBS
- Amazon EC2

### Other resources

- How to move Micrsoft SQL Server tempdb to instance/ephemeral disks on Amazon EC2
- Stripe Windows Ephemeral Disks at Launch
- How much memory does my SQL Server actually need?
- SQL Server Wait Statistics (or please tell me where it hurts…)
- Connection Timeouts in Multi-subnet Availability Group
- Plan cache and optimizing for adhoc workloads
- RAM, virtual memory, pagefile, and memory management in Windows
- How to determine the appropriate page file size for 64-bit versions of Windows

# Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

| Change | Description | Date |
|---|---|---|
| Corrected information | Corrected the information about the cost threshold of parallelism property. Its value is measured by units of cost, not time. | December 4, 2023 |
| Updated guidance | Updated the sections on setting the NTFS allocation unit size, locking pages in memory, using task offload features, using striping, changing the cost threshold of parallelism, using trace flags, and using database autogrowth settings. | August 8, 2023 |
| Added guidance | Added information about using Network Load Balancer for legacy applications that can’t use the **MultiSubnetFailover** parameter. | November 11, 2022 |
| Fixed code | Corrected the PowerShell code in the section on initializing the instance store. | June 27, 2022 |
| Added new section | Added information about AWS Launch Wizard for SQL Server. | August 18, 2021 |

| | | |
|---|---|---|
| Initial publication | — | July 21, 2020 |

# AWS Prescriptive Guidance glossary

The following are commonly used terms in strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

## Numbers

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there’s no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

# A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include `SUM` and `MAX`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

# B

bad bot

A bot that is intended to disrupt or cause harm to individuals or organizations.

BCP

See business continuity planning.

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also endianness.

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

# C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

## D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

# E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.
- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

## F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with AWS.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the “2021-05-27 00:15:37” date into “2021”, “May”, “Thu”, and “15”, you can help the learning algorithm learn nuanced patterns associated with different data components.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

# G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries.

*Detective guardrails* detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub CSPM, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

# H

HA

See high availability.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

# I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

ITIL

See IT information library.

ITSM

See IT service management.

# L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

# M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed, and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization’s cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can use a microservices architecture. For more information, see Decomposing monoliths into microservices.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

# O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic PUT and DELETE requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

## P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

polyglot persistence

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store best adapted to their requirements. For more information, see Enabling data persistence in microservices.

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model’s responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

# Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

# R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that's used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it’s called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

# S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata.

The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

# T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

test environment

See environment.

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don’t know the target.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

# U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn’t provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

## V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

## W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

# Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.