Backup and recovery approaches on AWS

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Backup and recovery approaches on AWS

Table of Contents

Backup and recovery approaches on AWS

Khurram Nizami, Amazon Web Services (AWS)

June 2024 (document history)

This guide discusses how to implement backup and recovery approaches using Amazon Web Services (AWS) services for on-premises, cloud-native, and hybrid architectures. These approaches offer lower costs, higher scalability, and more durability to meet recovery time objective (RTO), recovery point objective (RPO), and compliance requirements.

This guide is intended for technical leaders who are responsible for protecting data in their corporate IT and cloud environments.

This guide covers different backup architectures (cloud-native applications, hybrid, and on-premises environments). It also covers associated Amazon Web Services (AWS) services that can be used to build scalable and reliable data-protection solutions for the non-immutable components of your architecture.

Another approach is to modernize your workloads to use immutable architectures, reducing the need for backup and recovery of components. AWS provides a number of services to implement immutable architectures and reduce the need for backup and recovery, including:

- Serverless with AWS Lambda
- Containers with Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), and AWS Fargate
- Amazon Machine Images (AMIs) with Amazon Elastic Compute Cloud (Amazon EC2)

As the growth of enterprise data accelerates, the task of protecting it becomes more challenging. Questions about the durability and scalability of backup approaches are commonplace, including this one: How does the cloud help meet my backup and restore needs?

This guide includes the following topics:

- Backup and recovery using Amazon S3
- Backup and recovery for Amazon EC2 with EBS volumes
- Backup and recovery from on-premises infrastructure to AWS
- Backup and recovery of applications from AWS to your data center
- Backup and recovery of cloud-native AWS services
- Backup and recovery for hybrid architectures
- Disaster recovery with AWS
- Cleaning up backups

Why use AWS as a data-protection platform?

AWS is a secure, high-performance, flexible, money-saving, and easy-to-use cloud computing platform. AWS takes care of the undifferentiated heavy lifting required to create, implement, and manage scalable backup and recovery solutions.

There are many advantages to using AWS as part of your data protection strategy:

- **Durability**: Amazon Simple Storage Service (Amazon S3) and S3 Glacier Deep Archive are designed for 99.999999999 percent (11 nines) of durability. Both platforms offer reliable backup of data, with object replication across at least three geographically dispersed Availability Zones. Many AWS services use Amazon S3 for storage and export/import operations. For example, Amazon Elastic Block Store (Amazon EBS) uses Amazon S3 for snapshot storage.
- **Security**: AWS provides a number of options for access control and data encryption while in-transit and at-rest.
- **Global infrastructure**: AWS services are available around the globe, so you can back up and store data in the Region that meets your compliance and workload requirements.
- **Compliance**: AWS infrastructure is certified for compliance with the following standards, so you can easily fit the backup solution into your existing compliance regimen:
  - Service Organization Controls (SOC)
  - Statement on Standards for Attestation Engagements (SSAE) 16
  - International Organization for Standardization (ISO) 27001
  - Payment Card Industry Data Security Standard (PCI DSS)
  - Health Insurance Portability and Accountability Act (HIPAA)
  - SEC1

  - Federal Risk and Authorization Management Program (FedRAMP)
- **Scalability**: With AWS, you don't have to worry about capacity. As your needs change, you can scale your consumption up or down without administrative overhead.
- **Lower total cost of ownership (TCO)**: The scale of AWS operations drives down service costs and helps lower the TCO of AWS services. AWS passes these cost savings on to customers through price drops.
- **Pay-as-you-go pricing**: Purchase AWS services as you need them and only for the period that you plan to use them. AWS pricing has no upfront fees, termination penalties, or long-term contracts.

Targeted business outcomes

The goal of this guide is to provide an overview of AWS services that you can use to support backup and recovery approaches for the following:

- On-premises architectures
- Cloud-native architectures
- Hybrid architectures
- AWS native services
- Disaster recovery (DR)

Best practices and considerations are covered along with an overview of services. This guide also provides you with the tradeoffs between using one approach over another for backup and recovery.

Choosing AWS services for data protection

AWS provides a number of storage and complementary services that can be used as part of your backup and recovery approach. These services can support both cloud-native and hybrid architectures. Different services are more effective for different use cases.

- Amazon S3 is suited for both hybrid and cloud-native use cases. It provides highly durable, general-purpose object storage solutions that are suitable for backing up individual files, servers, or an entire data center.
- AWS Storage Gateway is ideal for hybrid use cases. Storage Gateway uses the power of Amazon S3 for common on-premises backup and storage requirements. Your applications connect to the service through a virtual machine (VM) or hardware gateway appliance using the following standard storage protocols:
  - Network File System (NFS)
  - Server Message Block (SMB)
  - Internet Small Computer System Interface (iSCSI)

  The gateway bridges these common on-premises protocols to AWS storage services such as the following:
  - Amazon S3
  - S3 Glacier Deep Archive
  - Amazon EBS

  Storage Gateway makes it easier to provide elastic, high-performance storage for files, volumes, snapshots, and virtual tapes in AWS.
- AWS Backup is a fully managed backup service for centralizing and automating the backup of data across AWS services. Using AWS Backup, you can centrally configure backup policies and monitor backup activity for AWS resources, such as the following:
  - EBS volumes
  - EC2 instances (including Windows applications)
  - Amazon RDS and Amazon Aurora databases
  - DynamoDB tables
  - Amazon Neptune databases
  - Amazon DocumentDB (with MongoDB compatibility) databases

  - Amazon EFS file systems
  - Amazon FSx for Lustre file systems and Amazon FSx for Windows File Server file systems
  - Storage Gateway volumes

  The cost of AWS Backup is based on the storage that you consume, restore, and transfer in a month. For more information, see the AWS Backup pricing.

- AWS Elastic Disaster Recovery replicates your machines into a staging area subnet in your target AWS account and preferred Region. The staging area design reduces costs by using affordable storage and minimal compute resources to maintain ongoing replication. You can use Elastic Disaster Recovery for DR from on premises to the cloud and for cross-Region DR.
- AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. This includes how the resources are related to one another and how they were configured in the past. In this view, you can see how the resource configuration and relationships have changed over time.

  When you turn on AWS Config configuration recording for your AWS resources, you maintain a history of your resource relationships over time. This helps to identify and track AWS resource relationships (including deleted resources) for up to seven years. For example, AWS Config can track the relationship of an Amazon EBS snapshot volume and the EC2 instance to which the volume was attached.
- AWS Lambda can be used to programmatically define and automate your backup and recovery procedures for your workloads. You can use the AWS SDKs to interact with AWS services and their data. You can also use Amazon EventBridge to run your Lambda functions on a scheduled basis.

AWS services provide specific features for backup and restore. For each AWS service that you are using, consult the AWS documentation to determine the backup, restore, and data protection features provided by the service. You can use the AWS Command Line Interface (AWS CLI), AWS SDKs, and API operations to automate the AWS service–specific features for data backup and recovery.

Designing a backup and recovery solution

When developing a comprehensive strategy for backing up and restoring data, you must first identify possible failure or disaster situations and their potential business impact. In some industries, you must consider regulatory requirements for data security, privacy, and records retention.

Backup and recovery processes should include the appropriate level of granularity to meet recovery time objective (RTO) and recovery point objective (RPO) for the workload and its supporting business processes, including the following:

- File-level recovery (for example, configuration files for an application)
- Application data–level recovery (for example, a specific database within MySQL)
- Application-level recovery (for example, a specific web server application version)
- Amazon EC2 volume-level recovery (for example, an EBS volume)
- EC2 instance-level recovery. (for example, an EC2 instance)
- Managed service recovery (for example, a DynamoDB table)

Be sure to consider all the recovery requirements for your solution and the data dependencies between various components in your architecture. To facilitate a successful restore process, coordinate the backup and recovery between various components in your architecture.

The following topics describe backup and recovery approaches based on the organization of your infrastructure. IT infrastructure can broadly be categorized as on-premises, hybrid, or cloud native.

Backup and recovery using AWS Backup

AWS Backup is a fully managed backup service centralizing and automating the backup of data across AWS services. AWS Backup provides an orchestration layer that integrates Amazon CloudWatch, AWS CloudTrail, AWS Identity and Access Management (IAM), AWS Organizations, and other services. This centralized, AWS Cloud native solution provides global backup capabilities that can help you achieve your disaster recovery and compliance requirements. Using AWS Backup, you can centrally configure backup policies and monitor backup activity for AWS resources.

AWS Backup is an ideal solution for implementing standard backup plans for your AWS resources across your AWS accounts and Regions. Because AWS Backup supports multiple AWS resource types, it makes it easier to maintain and implement a backup strategy for workloads using multiple AWS resources that need to be backed up collectively. AWS Backup also enables you to collectively monitor a backup and restore operation that involves multiple AWS resources.

If you have compliance and audit requirements, you can use the AWS Backup Audit Manager feature to create audit frameworks and reports to support your compliance requirements. The AWS Backup Vault Lock feature also supports compliance requirements by enforcing a write-once, read-many (WORM) configuration for all your backups stored in an backup vault in AWS Backup.

A key differentiator for AWS Backup is support for Organizations. Using this support, you can define and manage backup policies at the organization or organizational unit level and automatically have those policies implemented for each related AWS account and Region. As you onboard new AWS accounts and Regions, you don't have to define and manage backup plans separately.

AWS Backup can make it easier for you to implement an organization-wide backup policy by using tags. You can create separate backup plans that each have unique frequency and retention settings and then create unique key-value pair tags that select the resources to include for backup.

For example, you could create a daily backup plan that starts a backup at 05:00 UTC on a daily basis and has a 35-day retention policy. This backup plan can include a backup resource assignment that specifies that any supported AWS resource with the tag key **backup** and tag value **daily** will be backed up according to this plan. Additionally, you could create a monthly backup plan that starts at 05:00 UTC on the first day of each month and has a 366-day retention policy. This backup plan can include a backup resource assignment that specifies that any supported AWS resource with the tag key **backup** and tag value **monthly** will be backed up according to this plan.

You can then use tag policies and the required-tags AWS Config rule to ensure that all your AWS supported resources have this tag key and one of these tag values. This approach can help you consistently implement and maintain a standard backup approach in AWS for supported AWS Backup resources. You can extend this approach to standardize backups for your applications and architectural layers that have different recovery point objective (RPO) requirements.

We recommend taking steps to secure your backup vault. For example, you can implement an Organizations service control policy (SCP) that prevents your backup vault from being deleted or from being shared with unintended AWS accounts. For more details and other important security considerations, review the Top 10 security best practices for securing backups in AWS blog post.

AWS Backup can simplify implementation of your disaster recovery (DR) plan for AWS because it supports multiple AWS resources that can be addressed collectively. For example, you can implement cross-Region and cross-account backup for most of the AWS resource types supported by AWS Backup. Cross-account backup improves backup security because a copy is available in a separate account. Cross-Region backup improves availability because the backups are available in more than one Region. For details about supported AWS resource types, see the Feature availability by resource table.

You can use the example Backup and Recovery with AWS Backup open-source solution to implement an infrastructure as code (IaC) and continuous integration and continuous delivery (CI/CD) approach to managing backups for your AWS Organizations organization. This solution includes custom features, such as automatically reapplying AWS tags on restored AWS resources as well as establishing a secondary backup vault in a separate account and Region for DR purposes.

Backup and recovery using Amazon S3

You can use Amazon Simple Storage Service (Amazon S3) to store and retrieve any amount of data, at any time. You can use Amazon S3 as your durable store for your application data and file-level backup and restore processes. For example, you can copy your database backups from a database instance to Amazon S3 with a backup script using the AWS CLI or AWS SDKs.

AWS services use Amazon S3 for highly durable and reliable storage, as in the following examples:

- Amazon EC2 uses Amazon S3 to store Amazon EBS snapshots for EBS volumes and for EC2 instance stores.
- Storage Gateway integrates with Amazon S3 to provide on-premises environments with Amazon S3 backed file shares, volumes, and tape libraries.
- Amazon RDS uses Amazon S3 for database snapshots.

Many third-party backup solutions also use Amazon S3. For example, Arcserve Unified Data Protection supports Amazon S3 for durable backup of on-premises and cloud-native servers.

You can use the Amazon S3 integrated features of these services to simplify your backup and recovery approach. At the same time, you can benefit from the high durability and availability provided by Amazon S3.

Amazon S3 stores data as objects within resources called buckets. You can store as many objects as you want in a bucket. You can write, read, and delete objects in your bucket with fine-grained access control. Single objects can be up to 5 TB in size.

Using Amazon S3 storage classes to reduce backup data storage costs

Amazon S3 offers multiple storage classes for use in on-premises, hybrid, and cloud-native architectures. All storage classes provide scalable capacity that requires no volume or media management as your backup datasets grow. The pay-for-what-you-use model and low cost per GB/month make Amazon S3 storage classes a fit for a broad range of data-protection use cases. Amazon S3 storage classes are designed for different use cases, including the following categories:

- Frequent access storage classes for general-purpose storage of frequently accessed data (for example, configuration files, unplanned backups, daily backups). This includes the S3 Standard storage class, which is the default for all Amazon S3 objects.
- Infrequent access storage classes for long-lived, but infrequently accessed data (for example, monthly backups). This includes the S3 Standard-IA storage class. IA stands for *infrequent access*.
- S3 Glacier storage classes for extremely long-lived data that rarely needs to be accessed (for example, yearly backups). This includes S3 Glacier Deep Archive, which provides the lowest-cost storage on AWS.

For backups with unknown or changing access patterns, you can use the S3 Intelligent-Tiering storage class. S3 Intelligent-Tiering automatically transitions objects to the most cost-effective tier based on how many days ago an object was last accessed.

> **Note**
>
> Some storage classes have a minimum duration charge. For details, see Amazon S3 pricing, and use the web page search to find `duration`.

Amazon S3 offers lifecycle policies that you can configure to manage your data throughout its lifecycle. After a policy is set, your data will be automatically migrated to the appropriate storage class without any changes to your application. For more information, see the Amazon S3 object lifecycle management documentation.

To reduce your costs for backup, use a tiered storage class approach based on your recovery time objective (RTO) and recovery point objective (RPO), as in the following example:

- Daily backups for the past 2 weeks using S3 Standard
- Weekly backups for the past 3 months using S3 Standard-IA
- Quarterly backups for the past year on S3 Glacier Flexible Retrieval
- Yearly backups for the past 5 years on S3 Glacier Deep Archive
- Backups deleted from S3 Glacier Deep Archive after the 5-year mark

Creating standard S3 buckets for backup and archive

You can create a standard S3 bucket for backup and archive with your corporation's backup and retention policy implemented through S3 lifecycle policies. Cost allocation tagging and reporting for AWS billing is based on the tags assigned at the bucket level. If cost allocation is important, create separate backup and archive S3 buckets for each project or business unit so that you can allocate costs accordingly.

Your backup scripts and applications can use the backup and archive S3 bucket that you create to store point-in-time snapshots for application and workload data. You can create a standard S3 prefix to help you organize your point-in-time data snapshots. For example, if you create hourly backups, consider using a backup prefix such as `YYYY/MM/DD/HH/<WorkloadName>/<files...>`. By doing this, you can quickly retrieve your point-in-time backups manually or programmatically.

Using Amazon S3 versioning to automatically maintain rollback history

You can enable S3 object versioning to maintain a history of object changes, including the ability to revert to a previous version. This is useful for configuration files and other objects that might change more frequently than your point-in-time backup schedule. It's also useful for files that must be reverted individually.

Using Amazon S3 to back up and recover customized configuration files for AMIs

Amazon S3 with object versioning can become your system of record for your workload configuration and option files. For example, you might use a standard AWS Marketplace Amazon EC2 image that is maintained by an ISV. This image might contain software whose configuration is maintained in a number of configuration files. You can maintain your customized configuration files in Amazon S3. When your instance is launched, you can copy these configuration files to your instance as a part of your instance user data. When you apply this approach, you don't need to customize and recreate an AMI to use an updated version.

Using Amazon S3 in your custom backup and restore process

Amazon S3 provides a general-purpose backup store that you can quickly integrate into your existing custom backup processes. You can use the AWS CLI, AWS SDKs, and API operations to integrate your backup and restore scripts and processes that use Amazon S3. For example, you might have a database backup script that performs nightly database exports. You can customize this script to copy your nightly backups to Amazon S3 for offsite storage. See the Batch upload files to the cloud tutorial for an overview of how to do this.

You can take a similar approach for exporting and backing up data for different applications based on their individual RPO. Additionally, you can use AWS Systems Manager to run your backup scripts on your managed instances. Systems Manager provides automation, access control, scheduling, logging, and notification for your individual backup processes.

Securing backup data in Amazon S3

Data security is a universal concern, and AWS takes security very seriously. Security is the foundation of every AWS service. Amazon S3 provides capabilities for access control and encryption both at rest and in transit. All Amazon S3 endpoints support SSL/TLS for encrypting data in transit. You can set up encryption for objects at rest by doing the following:

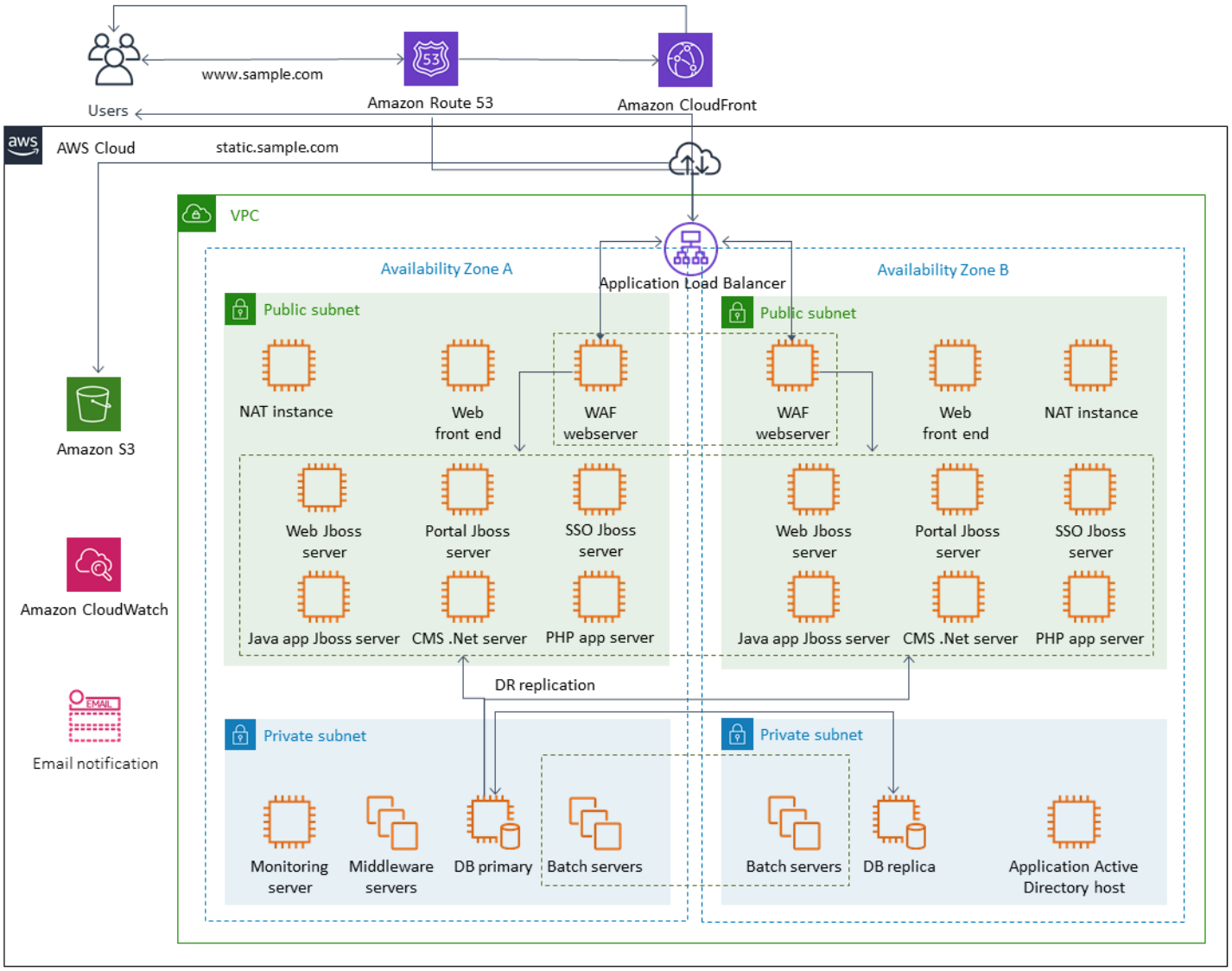
- Using server-side encryption with Amazon S3 managed encryption keys (default)
- Using server-side encryption with AWS Key Management Service (AWS KMS) keys stored in AWS KMS
- Using client-side encryption

You can use AWS Identity and Access Management (IAM) to control access to S3 objects. IAM provides control over permissions for individual objects and specific prefix paths within an S3 bucket. You can audit access to S3 objects by using object-level logging with AWS CloudTrail.

Backup and recovery for Amazon EC2 with EBS volumes

AWS provides multiple methods to back up your Amazon EC2 instances. This section covers different aspects of backing up Amazon Elastic Block Store (Amazon EBS) volumes or instance store volumes for storage. Consider AWS Backup as your first choice for managing backups on AWS if it meets your requirements. Remember that backups are good only if they can be restored to the function for which they were intended. The restore and recovery function should be regularly tested to confirm this.

The solution architecture in the following diagram describes a workload environment that exists entirely on AWS with the majority of the architecture based on Amazon EC2. As the following figure shows, the scenario includes web servers, application servers, monitoring servers, databases, Active Directory, and disaster recovery (DR) replication.



AWS provides many fully featured services for many of the Amazon EC2 servers represented in this architecture to perform the undifferentiated work of creating, provisioning, backing up, restoring, and optimizing the instances and storage. Consider whether these services make sense in your architecture to reduce complexity and management. AWS also provides services to improve the availability of your Amazon EC2–based architectures. In particular, consider Amazon EC2 Auto Scaling and Elastic Load Balancing to complement your workloads on Amazon EC2. Using these services can improve the availability and fault tolerance of your architecture and help you to restore impaired instances with minimal user impact.

EC2 instances primarily use Amazon EBS volumes for persistent storage. Amazon EBS provides a number of features for backup and recovery that are covered in detail in this section.

Topics

- Amazon EC2 backup and recovery with snapshots and AMIs
- Creating EBS volume backups with AMIs and EBS snapshots
- Restoring an Amazon EBS volume or an EC2 instance

Amazon EC2 backup and recovery with snapshots and AMIs

Consider whether you need to create a full backup of an EC2 instance with an Amazon Machine Image (AMI) or take a snapshot of an individual volume.

Using AMIs or Amazon EBS snapshots for backups

An AMI includes the following:

- One or more snapshots. Instance-store-backed AMIs include a template for the root volume of the instance (for example, an operating system, an application server, and applications).
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

> **Note**
>
> In most cases, AMIs for Windows, Red Hat, SUSE, and SQL Server require correct licensing information to be present on the AMI. For more information, see Understand AMI billing information. When creating an AMI from a snapshot, the `RegisterImage` operation derives the correct billing information from the snapshot's metadata, but this requires the appropriate metadata to be present. To verify if the correct billing information was applied, check the **Platform details** field on the new AMI. If the field is empty or doesn't match the expected operating system code (for example, Windows, Red Hat, SUSE, or SQL), the AMI creation was unsuccessful, and you should discard the AMI and follow the instructions in Create an AMI from an instance.

You can use AMIs to launch new instances with preconfigured software and data. You can create AMIs when you want to establish a baseline, which is a reusable configuration for launching more instances. When you create an AMI of an existing EC2 instance, a snapshot is taken for all the volumes that are attached to the instance. The snapshot includes the device mappings.

You can't use snapshots to launch a new instance, but you can use them to replace volumes on an existing instance. If you experience data corruption or a volume failure, you can create a volume from a snapshot that you have taken and replace the old volume. You can also use snapshots to provision new volumes and attach them during a new instance launch.

If you are using platform and application AMIs maintained and published by AWS or from the AWS Marketplace, consider maintaining separate volumes for your data. You can back up your data volumes as snapshots that are separate from the operating system and application volumes. Then use the data volume snapshots with newly updated AMIs published by AWS or from the AWS Marketplace. This approach requires careful testing and planning to back up and restore all custom data, including configuration information, on the newly published AMIs.

The restore process is affected by your choice between AMI backups or snapshot backups. If you create AMIs to serve as instance backups, you must launch an EC2 instance from the AMI as a part of your restore process. You might also need to shut down the existing instance to avoid potential collisions. An example of a potential collision is security identifiers (SIDs) for domain-joined Windows instances. The restore process for snapshots might require you to detach the existing volume and attach the newly restored volume. Or you might need to make a configuration change to point your applications to the newly attached volume.

AWS Backup supports both instance-level backups as AMIs and volume-level backups as separate snapshots:

- For a full backup of all EBS volumes on the instance, create an AMI of the EC2 instance. When you want to roll back, use the launch instance wizard to create an instance. In the instance launch wizard, choose **My AMIs**.
- To back up an individual volume, create a snapshot. To restore the snapshot, see Create a volume from a snapshot. You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI).

The cost of an instance AMI is the storage of all the volumes on the instance, but not the metadata. The cost for an EBS snapshot is the storage of the individual volume. For more information about volume storage costs, see the Amazon EBS pricing page.

Server volumes

EBS volumes are the primary persistent storage option for Amazon EC2. You can use this block storage for structured data, such as databases, or unstructured data, such as files in a file system on a volume.

EBS volumes are placed in a specific Availability Zone. The volumes are replicated across multiple servers to prevent the loss of data from the failure of any single component. Failure refers to a complete or partial loss of the volume, depending on the size and performance of the volume.

EBS volumes are designed for an annual failure rate (AFR) of 0.1-0.2 percent. This makes EBS volumes 20 times more reliable than typical commodity disk drives, which fail with an AFR of around 4 percent. For example, if you have 1,000 EBS volumes running for 1 year, you should expect one or two volumes will have a failure.

Amazon EBS also supports a snapshot feature for taking point-in-time backups of your data. All EBS volume types offer durable snapshot capabilities and are designed for 99.999 percent availability. For more information, see the Amazon Compute Service Level Agreement.

Amazon EBS provides the ability to create snapshots (backups) of any EBS volume. A snapshot is a base feature for creating backups of your EBS volumes. A snapshot takes a copy of the EBS volume and places it in Amazon S3, where it is stored redundantly in multiple Availability Zones. The initial snapshot is a full copy of the volume; ongoing snapshots store incremental block-level changes only. See the Amazon EBS documentation for details on how to create Amazon EBS snapshots.

You can perform a restore operation, delete a snapshot, or update the snapshot metadata, such as tags, associated with the snapshot from the Amazon EC2 console in the same Region that you took the snapshot.

Restoring a snapshot creates a new Amazon EBS volume with full volume data. If you need only a partial restore, you can attach the volume to the running instance under a different device name. Then mount it, and use operating system copy commands to copy the data from the backup volume to the production volume.

Amazon EBS snapshots can also be copied between AWS Regions by using the Amazon EBS snapshot copy capability, as described in the Amazon EBS documentation. You can use this feature to store your backup in another Region without having to manage the underlying replication technology.

Establishing separate server volumes

You may already use a standard set of separate volumes for the operating system, logs, applications, and data. By establishing separate server volumes, you can reduce the scope of impact when there are application or platform failures caused by disk space exhaustion. This risk is usually greater with physical hard drives, because you don't have the flexibility to expand volumes quickly. With physical drives, you must purchase the new drives, back up the data, and then restore the data on the new drives. With AWS, this risk is greatly reduced because you can use Amazon EBS to expand your provisioned volumes. For more information, see the AWS documentation.

Maintain separate volumes for application data, user data, logs, and swap files so that you can use separate backup and restore policies for these resources. By separating volumes for your data, you can also use different volume types based on the performance and storage requirements for the data. You can then optimize and fine-tune your costs for different workloads.

Considerations for instance store volumes

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance stores are ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content. They are also preferable for data that are replicated across a fleet of instances, such as a load balanced pool of web servers.

The data in an instance store persists only during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data in the instance store is lost under any of the following circumstances.

- The underlying drive fails.
- The instance stops.
- The instance terminates.

Therefore, do not rely on an instance store for valuable, long-term data. Instead, use more durable data storage, such as Amazon S3, Amazon EBS, or Amazon EFS.

A common strategy with instance store volumes is to persist necessary data to Amazon S3 regularly as needed, based on the recovery point objective (RPO) and recovery time objective (RTO). You can then download the data from Amazon S3 to your instance store when a new

instance is launched. You can also upload the data to Amazon S3 before an instance is stopped. For persistence, create an EBS volume, attach it to your instance, and copy the data from the instance store volume to the EBS volume on a periodic basis. For more information, see the AWS Knowledge Center.

Tagging and enforcing standards for EBS snapshots and AMIs

Tagging all your AWS resources is an important practice for cost allocation, auditing, troubleshooting, and notification. Tagging is important for EBS volumes so that the pertinent information required to manage and restore volumes is present. Tags are not automatically copied from EC2 instances to AMIs or from source volumes to snapshots. Make sure that your backup process includes the relevant tags from these sources. This helps you to set the snapshot metadata, such as access policies, attachment information, and cost allocation, to use these backups in the future. For more information on tagging your AWS resources, refer to the tagging best practices technical paper.

In addition to the tags you use for all AWS resources, use the following backup-specific tags:

- Source instance ID
- Source volume ID (for snapshots)
- Recovery point description

You can enforce tagging policies by using AWS Config rules and IAM permissions. IAM supports enforced tag usage, so you can write IAM policies that mandate the use of specific tags when acting on Amazon EBS snapshots. If a `CreateSnapshot` operation is attempted without the tags defined in the IAM permissions policy granting rights, the snapshot creation fails with access denied. For more information, see the blog post on tagging Amazon EBS snapshots on creation and implementing stronger security policies.

You can use AWS Config rules to evaluate the configuration settings of your AWS resources automatically. To help you get started, AWS Config provides customizable, predefined rules called managed rules. You can also create your own custom rules. While AWS Config continuously tracks configuration changes among your resources, it checks whether these changes violate any of the conditions in your rules. If a resource violates a rule, AWS Config flags the resource and the rule as *noncompliant*. Note that the required-tags managed rule does not currently support snapshots and AMIs.

Creating EBS volume backups with AMIs and EBS snapshots

AWS provides a wealth of options for creating and managing AMIs and snapshots. You can use the approach that meets your needs. A common issue that many customers face is managing the snapshot lifecycle and clearly aligning snapshots by purpose, retention policy, etc. Without proper tagging, there is a risk that snapshots might be deleted accidentally or as part of an automated cleanup process. You might also end up paying for obsolete snapshots that are retained because there is no clear understanding whether they are still needed.

Preparing an EBS volume before creating a snapshot or AMI

Before you take a snapshot or create an AMI, make the necessary preparations to your EBS volume. Creating an AMI results in a new snapshot for each EBS volume that is attached to the instance, so these preparations also apply to AMIs.

You can take a snapshot of an attached EBS volume that is in use by a powered-on EC2 instance. However, snapshots capture only data that has been written to your EBS volume at the time the snapshot command is issued. This might exclude any data that has been cached by applications or the operating system. A best practice is to have the system in a state where it is not performing any I/O. Ideally, the machine isn't accepting traffic and is in a stopped state, but this is rare as 24/7 IT operations become the norm. If you can flush any data from system memory to the disk being used by your applications and pause any file writes to the volume long enough to take a snapshot, your snapshot should be complete.

To make a clean backup, you must quiesce the database or file system. The way in which you do this depends on your database or file system.

The process for a database is as follows:

1. If possible, put the database into hot backup mode.
2. Run the Amazon EBS snapshot commands.
3. Take the database out of hot backup mode or, if using a read replica, terminate the read replica instance.

The process for a file system is similar, but it depends on the capabilities of the operating system or file system. For example, XFS is a file system that can flush its data for a consistent backup. For more information, see xfs_freeze. Alternatively, you can facilitate this process by using a logical volume manager that supports the freezing of I/O.

However, if you can't flush or pause all file writes to the volume, do the following:

1. Unmount the volume from the operating system.
2. Issue the snapshot command.
3. Remount the volume to achieve a consistent and complete snapshot. You can remount and use your volume while the snapshot status is pending.

The snapshot process continues in the background and snapshot creation is fast and captures a point in time. The volumes that you're backing up are unmounted for only a matter of seconds. You can schedule a small backup window where an outage is expected and handled by clients gracefully.

When you create a snapshot for an EBS volume that serves as a root device, stop the instance before you take the snapshot. Windows provides the Volume Shadow Copy Service (VSS) to help create application-consistent snapshots. AWS provides a Systems Manager document that you can run to take image-level backups of VSS-aware applications. The snapshots include data from pending transactions between these applications and the disk. You don't have to shut down your instances or disconnect them when you back up all attached volumes. For more information, see the AWS documentation.

Note

If you are creating a Windows AMI so that you can deploy another similar instance, use EC2Config or EC2Launch to Sysprep your instance. Then create an AMI from the stopped instance. Sysprep removes unique information from the Amazon EC2 Windows instance, including the SIDs, computer name, and drivers. Duplicate SIDs can cause issues with Active Directory, Windows Server Update Services (WSUS), login issues, Windows volume key activation, Microsoft Office, and third-party products. Do not use Sysprep with your instance if your AMI is for backup purposes and you want to restore the same instance with all its unique information intact.

Creating EBS volume snapshots manually from the console

Create snapshots of the appropriate volumes or the entire instance before you make any major changes that have not been fully tested on the instance. For example, you might want to create a snapshot before you upgrade or patch application or system software on your instance.

You can create a snapshot manually from the console. On the Amazon EC2 console, on the **Elastic Block Store Volumes** page, select the volume that you want to back up. Then on the **Actions** menu, choose **Create Snapshot**. You can search for volumes that are attached to a specific instance by entering the instance ID in the filter box.

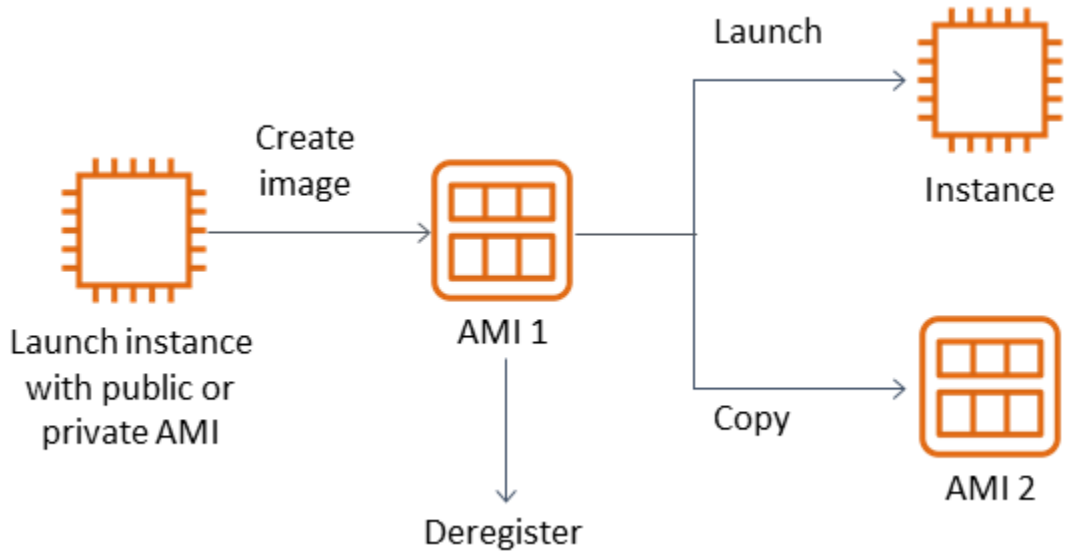
Enter a description and add the appropriate tags. Add a `Name` tag to make it easier to find the volume later. Add any other appropriate tags based on your tagging strategy.

Creating AMIs

An AMI provides the information that is required to launch an instance. The AMI includes the root volume and snapshots of the EBS volumes attached to the instance when the image was created. You can't launch new instances from EBS snapshots alone; you must launch new instances from an AMI.

When you create an AMI, it is created in the account and Region that you are using. The AMI creation process creates Amazon EBS snapshots for each volume attached to the instance, and the AMI refers to these Amazon EBS snapshots. These snapshots reside in Amazon S3 and are highly durable.

After you create an AMI of your EC2 instance, you can use the AMI to re-create the instance or launch more copies of the instance. You can also copy AMIs from one Region to another for application migration or DR.



An AMI must be created from an EC2 instance unless you are migrating a virtual machine, such as a VMWARE virtual machine, to AWS. To create an AMI from the Amazon EC2 console, select the instance, choose **Actions**, choose **Image**, and then choose **Create Image**.

Amazon Data Lifecycle Manager

To automate the creation, retention, and deletion of Amazon EBS snapshots, you can use Amazon Data Lifecycle Manager. Automating snapshot management helps you to do the following:

- Protect valuable data by enforcing a regular backup schedule.
- Retain backups as required by auditors or internal compliance.
- Reduce storage costs by deleting outdated backups.

Using Amazon Data Lifecycle Manager, you can automate the snapshot management process for EC2 instances (and their attached EBS volumes) or separate EBS volumes. It supports options such as cross-Region copy, so you can copy snapshots automatically to other AWS Regions. Copying snapshots to alternative Regions is one approach to support DR efforts and restore options in an alternative Region. You can also use Amazon Data Lifecycle Manager to create a snapshot lifecycle policy that supports fast snapshot restore.

Amazon Data Lifecycle Manager is an included feature of Amazon EC2 and Amazon EBS. There is no charge for Amazon Data Lifecycle Manager.

AWS Backup

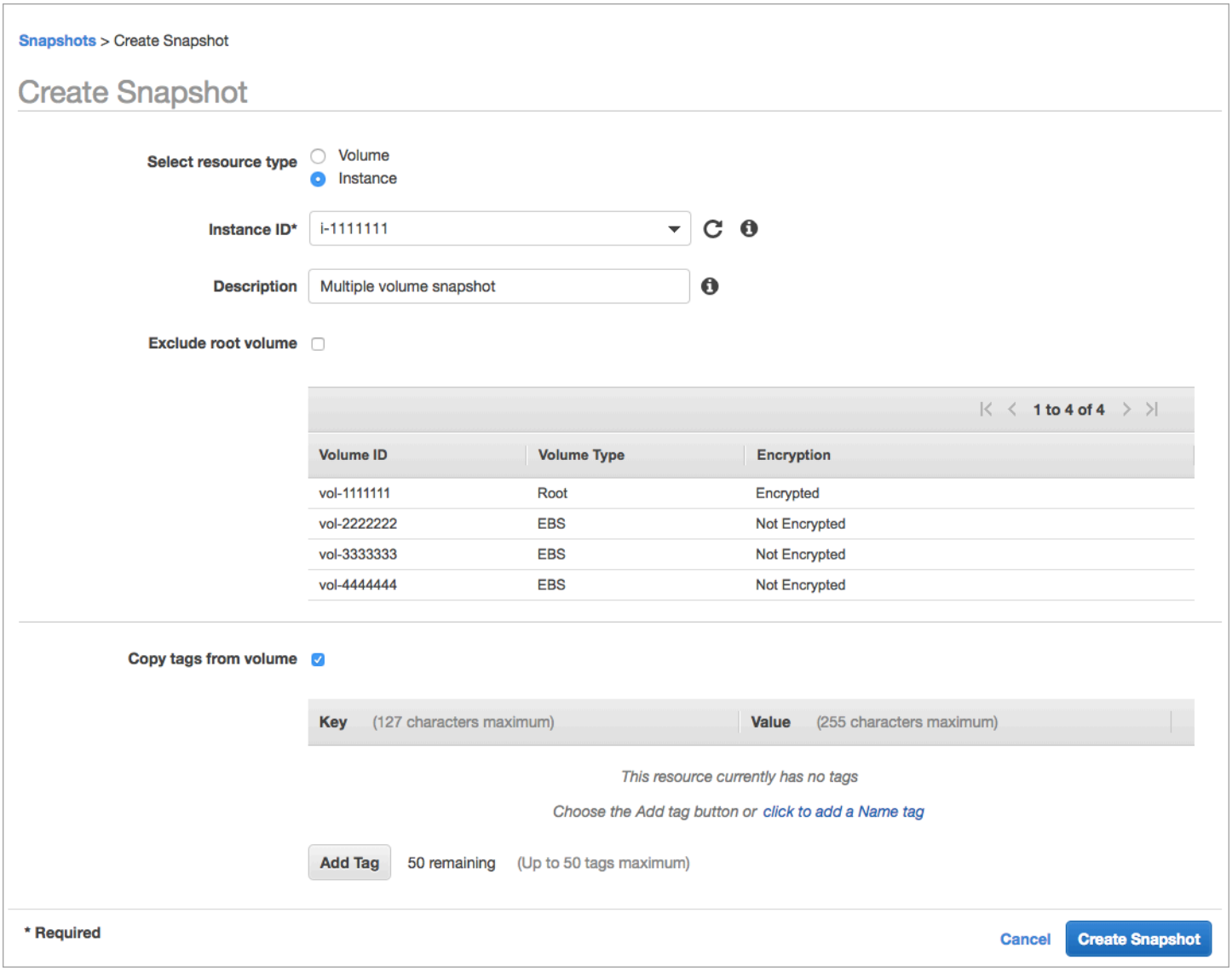
AWS Backup is unique from Amazon Data Lifecycle Manager because you can create a backup plan that includes resources across multiple AWS services. You can coordinate your backup to cover the resources you are using together rather than coordinating the backups of the resources individually.

AWS Backup also includes the concept of backup vaults, which can restrict access to the recovery points for your completed backups. Restore operations can be initiated from AWS Backup rather than proceeding to each individual resource and restoring the backup created. AWS Backup also includes a host of additional features, such as audit management and reporting. For more information, see the Backup and recovery using AWS Backup section of this guide.

Performing multi-volume backups

If you want to back up the data on the EBS volumes in a RAID array using snapshots, the snapshots must be consistent. This is because the snapshots of these volumes are created independently. Restoring EBS volumes in a RAID array from snapshots that are out of sync degrades the integrity of the array.

To create a consistent set of snapshots for your RAID array, use the CreateSnapshots API operation, or log in to the Amazon EC2 console and choose **Elastic Block Store**, **Snapshots**, **Create Snapshot**.



Snapshots of instances that have multiple volumes attached in a RAID configuration are taken as a multi-volume snapshot, collectively. Multi-volume snapshots provide point-in-time, data-coordinated, and crash-consistent snapshots across multiple EBS volumes attached to an EC2

instance. You do not have to stop your instance to coordinate between volumes to achieve consistency because snapshots are automatically taken across multiple EBS volumes. After the snapshot for the volumes is initiated (usually a second or two), the file system can continue its operations.

After the snapshots are created, each snapshot is treated as an individual snapshot. You can perform all snapshot operations, such as restore, delete, and cross-Region and account copy, as you would with a single-volume snapshot. You can also tag your multi-volume snapshots as you would a single-volume snapshot. We recommend that you tag your multi-volume snapshots to manage them collectively during restore, copy, or retention. For more information, see the AWS documentation.

You can also perform these backups from a logical volume manager or a file system–level backup. In these cases, using a traditional backup agent enables the data to be backed up over the network. A number of agent-based backup solutions are available on the internet and in the AWS Marketplace.

An alternative approach is to create a replica of the primary system volumes that exist on a single large volume. This simplifies the backup process, because only one large volume must be backed up, and the backup does not take place on the primary system. However, first determine whether the single volume can perform sufficiently during the backup and whether the maximum volume size is appropriate for the application.

Protecting your Amazon EC2 backups

It is important to consider the security of your backups and to prevent accidental or malicious deletion of your backups. You can use a number of approaches collectively to accomplish this. To prevent the loss of your critical backups due to a security breach, we recommend that you copy your backups to another AWS account. If you have multiple AWS accounts, you can designate a separate account as your archive account to which all the other accounts can copy backups. For example, you can accomplish this with a cross-account backup in AWS Backup.

Your disaster recovery plan might also require you to be able to reproduce EC2 instances in another AWS Region in case of a regional failure. You can support this goal by copying your backups to another Region within the same account. This can provide an additional layer of accidental deletion protection as well as support disaster recovery (DR) objectives. AWS Backup provides support for cross-Region backups.

Consider blocking IAM permissions to the ec2:DeleteSnapshot and ec2:DeregisterImage actions. Instead, you can let your retention policies and methods manage the lifecycle of EBS snapshots and Amazon EC2 AMIs. Blocking delete actions is one way to implement a write-once, read-many (WORM) strategy for your EBS snapshots. You can also use AWS Backup Vault Lock, which provides support for EBS snapshots and other AWS resources.

Additionally, consider blocking the ability for users to share AMIs and EBS snapshots by blocking the ec2:ModifyImageAttribute and ec2:ModifySnapshotAttribute IAM actions. This will prevent your AMIs and snapshots from being shared with AWS accounts that are external to your organization. If you are using AWS Backup, limit users from performing similar operations on backup vaults. For more information, see the *AWS Backup* section of this guide.

Amazon EBS includes a Recycle Bin feature that can help you restore accidentally deleted EBS snapshots. If you allow your users to delete snapshots, turn this feature on so that needed snapshots aren't permanently deleted. Users should be particularly careful about deleting multiple snapshots, because the Amazon EC2 console allows you to select multiple snapshots and delete them in one operation. Additionally, be careful when you use cleanup scripts and automation so that you don't unintentionally delete snapshots you need. The Recycle Bin feature helps provide protection from these types of situations.

Archiving EBS snapshots

Archiving your EBS snapshots can be a cost-effective method for keeping a copy of a volume for reference purposes that you don't intend to restore for 90 or more days. This can be a good intermediate step before permanently deleting all related snapshots for an EBS volume. For example, you might consider archiving snapshots as an end-of-lifecycle step for EBS volumes that are no longer used. Archiving rather than deleting can also be a more cost effective method of deletion retention instead of using the Recycle Bin.

Automating snapshot and AMI creation with Systems Manager, the AWS CLI, and the AWS SDKs

Your backup approach might require operations before and after a snapshot or AMI is created. For example, you might need to stop and start services to quiesce the file system. Or you might need to stop and start your instance during AMI creation. You might also need to create backups of multiple components in your architecture collectively, each with its own pre-creation and post-creation steps.

You can reduce your maintenance window times for your backups by automating your process and verifying that your backup process is consistently applied. To automate your custom pre-creation and post-creation operations, script your backup process by using the AWS CLI and the SDK.

Your automation can be defined in a Systems Manager runbook that can be run on demand or during a Systems Manager maintenance window. You can grant your users access to run Systems Manager runbooks without the need to grant them permissions to Amazon EC2 disruptive commands. This can also help you verify that your backup process and tags are applied consistently by your users. You can use the AWS-CreateSnapshot and AWS-CreateImage runbooks for creating snapshots and AMIs, or you can grant other users permissions to use them. Systems Manager also includes the AWS-UpdateLinuxAmi and AWS-UpdateWindowsAmi runbooks to automate the AMI patching and AMI creation.

You can also use the AWS CLI and AWS Tools for Windows PowerShell to automate your snapshot and AMI creation process. You can use the aws ec2 create-snapshot AWS CLI command to create a snapshot of an EBS volume as one step in your automation. You can use the aws ec2 create-snapshots command to create crash-consistent, synchronized snapshots of all volumes that are attached to your EC2 instance.

You can use the AWS CLI to create new AMIs. You can use the aws ec2 register-image command to create a new image for your EC2 instance. To automate the shutdown, image creation, and restart of your instances, combine this command with the aws ec2 stop-instances and aws ec2 start-instances commands.

Restoring an Amazon EBS volume or an EC2 instance

If you need to restore only a single volume attached to an EC2 instance, you can restore that volume separately, detach the existing volume, and attach the restored volume to your EC2 instance. If you need to restore an entire EC2 instance, including all of its associated volumes, you must use an Amazon Machine Image (AMI) backup of your instance.

To reduce the recovery time and impact to dependent applications and processes, your restore process must consider the resource that it is replacing. For best results, regularly test your restore process in lower environments (for example, non-production) to verify that your process meets your recovery point objective (RPO) and recovery time objective (RTO) and that the restore process works as expected. Consider how the restore process will impact applications and services that depend on the instance you are restoring, and then coordinate the restore as necessary. Try to automate and test the restore process as much as possible to reduce the risk of your restore process failing or being implemented inconsistently.

If you use Elastic Load Balancing, with multiple instances servicing traffic, you can take a failed or impaired instance out of service. Then you can restore a new instance to replace it while the other instances continue to service traffic without disruption to users.

The following restore processes described are for instances that are not using Elastic Load Balancing:

- Restoring individual files and directories from EBS snapshots
- Restoring an EBS volume from an Amazon EBS snapshot
- Creating or restoring an EC2 instance from an EBS snapshot
- Restoring a running instance from an AMI

Restoring files and directories from EBS snapshots

EBS snapshots provide a point-in-time exact replica of the original volume that was used to create the snapshot. To restore individual files or directories, you must do the following:

1. First, restore the volume from the EBS snapshot that contains the files or directories.
2. Attach the volume to the EC2 instance to which you want to restore the files.
3. Copy the files from the restored volume to your EC2 instance volume.
4. Detach and delete the restored volume.

Restoring an EBS volume from an Amazon EBS snapshot

You can restore a volume attached to an existing EC2 instance by creating a volume from its snapshot and attaching it to your instance. You can use the console, the AWS CLI, or the API operations to create a volume from an existing snapshot. You can then mount the volume to the instance by using the operating system.

Note that data from an Amazon EBS snapshot is asynchronously loaded into an EBS volume. If an application accesses the volume where the data is not loaded, there is higher latency than normal while the data is loaded from Amazon S3. To avoid this impact for latency-sensitive applications, you have two options:

- You can initialize the EBS volume.

- For an additional charge, Amazon EBS supports fast snapshot restore, which eliminates the need initialize your volume.

If you are replacing a volume that must use the same mount point, unmount that volume so that you can mount the new volume in its place. To unmount the volume, first stop any processes that are using the volume. If you are replacing the root volume, you must stop the instance first before you can detach the root volume.

For example, follow these steps to restore a volume to an earlier point-in-time backup by using the console:

1. On the Amazon EC2 console, on the **Elastic Block Store** menu, choose **Snapshots**.
2. Search for the snapshot that you want to restore, and select it.
3. Choose **Actions**, and then choose **Create Volume**.
4. Create the new volume in the same Availability Zone as your EC2 instance.
5. On the Amazon EC2 console, select the instance.
6. In the instance details, make note of the device name that you want to replace in the **Root device** entry or **Block Devices** entries.
7. Attach the volume. The process differs for root volumes and non-root volumes.

   For root volumes:

   a. Stop the EC2 instance.

   b. On the **EC2 Elastic Block Store Volumes** menu, select the root volume that you want to replace.

   c. Choose **Actions**, and then choose **Detach Volume**.

   d. On the **EC2 Elastic Block Store Volumes** menu, select the new volume.

   e. Choose **Actions**, and then choose **Attach Volume**.

   f. Select the instance that you want to attach the volume to, and use the same device name that you noted earlier.

   For non-root volumes:

   a. On the **EC2 Elastic Block Store Volumes** menu, select the non-root volume that you want to replace.

   b. Choose **Actions**, and then choose **Detach Volume**.

c. Attach the new volume by choosing it on the **EC2 Elastic Block Store Volumes** menu and then choosing **Actions**, **Attach Volume**. Select the instance that you want to attach it to, and then select an available device name.

d. Using the operating system for the instance, unmount the existing volume, and then mount the new volume in its place.

   In Linux, you can use the `umount` command. In Windows, you can use a logical volume manager (LVM) such as the Disk Management system utility.

e. Detach any prior volumes that you may be replacing by choosing it on the **EC2 Elastic Block Store Volumes** menu and then choosing **Actions**, **Detach Volume**.

You can also use the AWS CLI in combination with operating system commands to automate these steps.

Creating or restoring an EC2 instance from an EBS snapshot

To create a backup that will be used to restore an entire EC2 instance, we recommend creating an Amazon Machine Image (AMI). AMIs capture machine information such as virtualization type. They also create snapshots for each volume that is attached to the EC2 instance, including their device mappings, so that they can be restored in the same configuration.

> **Note**
>
> In most cases, AMIs for Windows, Red Hat, SUSE, and SQL Server require correct licensing information to be present on the AMI. For more information, see Understand AMI billing information. When creating an AMI from a snapshot, the `RegisterImage` operation derives the correct billing information from the snapshot's metadata, but this requires the appropriate metadata to be present. To verify if the correct billing information was applied, check the **Platform details** field on the new AMI. If the field is empty or doesn't match the expected operating system code (for example, Windows, Red Hat, SUSE, or SQL), the AMI creation was unsuccessful, and you should discard the AMI and follow the instructions in Create an AMI from an instance.

If you must use an EBS snapshot to restore an instance, first create an AMI from an EBS snapshot that will become the root volume for your new EC2 instance:

1. On the Amazon EC2 console, on the **Elastic Block Store** menu, choose **Snapshots**.
2. Search for the snapshot that will be used to create the root volume for your new EC2 instance, and select it.
3. Choose **Actions**, and then choose **Create Image from Snapshot**.
4. Enter a name for your image (for example, `YYYYMMDD-restore-for-i-012345678998765de`), and choose the appropriate options for your new image.
5. (Windows, Red Hat, SUSE, and SQL Server only) To verify if the correct billing information was applied, check the **Platform details** field on the new AMI. If the field is empty or doesn't match the expected operating system code (for example, **Windows** or **Red Hat**), the AMI creation was unsuccessful, and you should discard the AMI and follow the instructions in Create an AMI from an instance.

After the image is created and available, you can launch a new EC2 instance that will use the EBS snapshot for the root volume.

Restoring a running instance from an AMI

You can bring up a new instance from your AMI backup to replace an existing, running instance. One approach is to stop the existing instance, keep it offline while you launch a new instance from your AMI, and perform any necessary updates. This approach reduces the risk of conflicts from both instances running simultaneously. It is an acceptable approach if the services that your instance provides are down or you are performing the restore during a maintenance window. After you test your new instance, you can reassign any Elastic IP addresses that were allocated to the old instance. Then you can update any Domain Name Service (DNS) records to point to the new instance.

However, if during a restore you must minimize the downtime of your in-service instance, consider launching and testing a new instance from your AMI backup. Then replace the existing instance with the new instance.

While both instances are running, you must prevent the new instance from causing any platform-level or application-level collisions. For example, you might run into problems with domain-joined Windows instances that are running with the same SIDs and computer name. You might encounter similar issues with network applications and services that require unique identifiers.

To prevent other servers and services from connecting to your new instance before it's ready, use security groups to temporarily block all inbound connections for your new instance except for your

own IP address for access and testing. You can also block outbound connections temporarily for the new instance to prevent services and applications from initiating any connections or updates to other resources. When the new instance is ready, stop the existing instance, start services and processes on the new instance, and then unblock any inbound or outbound network connections that you implemented.

Backup and recovery from on-premises infrastructure to AWS

You can use AWS for durable, offsite storage of your on-premises infrastructure backups. By using AWS storage services in this scenario, you can focus on backup and archiving tasks. You don't have to worry about storage infrastructure provisioning, scaling, or infrastructure capacity for your backup tasks.

Amazon S3 provides extensive API operations and SDKs for integrating into your new and existing backup and recovery approaches. This also gives backup software vendors ways to directly integrate their applications with AWS storage solutions.

In this scenario, backup and archive software that you are using in your on-premises infrastructure directly interfaces with AWS through the API operations. Because the backup software is AWS-aware, it backs up the data from the on-premises servers directly to Amazon S3.

If your existing backup software does not natively support the AWS Cloud, you can use Storage Gateway. A cloud storage service, Storage Gateway gives your on-premises systems access to scalable cloud storage. It supports open standard storage protocols that work with your existing applications while securely storing your data encrypted in Amazon S3. You can use Storage Gateway as a part of a backup and recovery approach for your on-premises block-based storage workloads.

Storage Gateway is helpful in hybrid scenarios where you want to transition to cloud-based storage for your backups. Storage Gateway also helps you reduce capital investments in on-premises storage. You deploy Storage Gateway as a VM or a dedicated hardware appliance. This guide focuses on how Storage Gateway applies to backup and recovery.

Storage Gateway provides three different options to satisfy different requirements:

- A file gateway for storing application data files and backup images as durable objects on Amazon S3 cloud storage using SMB-based or NFS-based access.
- A volume gateway for presenting cloud-based iSCSI block storage volumes to your on-premises applications. A volume gateway provides either a local cache or full volumes on premises while also storing full copies of your volumes in the AWS Cloud.

- A tape gateway for pointing trusted backup software at an on-premises storage gateway that, in turn, connects to Amazon S3. This option delivers the scale and durability of the cloud for safe, long-term retention without disrupting existing investments or processes.

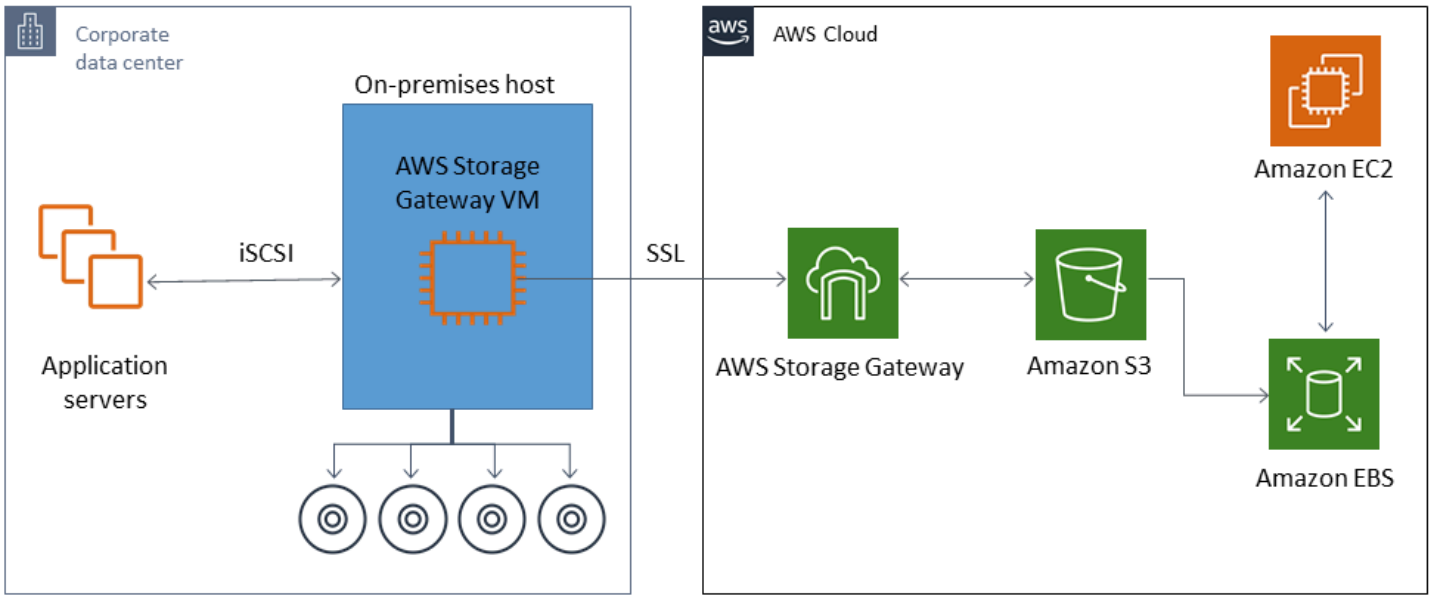
File gateway

Many organizations start their cloud journey by moving secondary and tertiary data, such as backups, to the cloud. A file gateway's SMB and NFS interface support provides a way for IT groups to transition backup jobs from existing on-premises backup systems to the cloud. Backup applications, native database tools, or scripts that can write to SMB or NFS can write to a file gateway. The file gateway stores the backups as Amazon S3 objects of up to 5 TiB in size. With an adequately sized local cache, recent backups can be used for fast on-site recoveries. Long-term retention needs are addressed by tiering backups to low-cost S3 Standard-Infrequent Access and S3 Glacier storage classes.

File gateway provides an on ramp for your block-based storage to Amazon S3 for highly durable offsite backups. It is especially useful for scenarios in which a recently backed up file must be restored quickly. Because a file gateway supports the SMB and NFS protocols, users can access files the same way they would access a network file share. You can also take advantage of Amazon S3 object versioning capabilities. Using object versioning, you can restore previous object versions for a file and then easily access them by using SMB or NFS.

Volume gateway

A volume gateway enables you to provision cloud-based iSCSI block storage volumes for your on-premises servers. The volume gateway stores your volume data to Amazon S3 for durable, scalable cloud-based offsite storage. A volume gateway facilitates taking full point-in-time snapshots of your volumes and storing them in the cloud as Amazon EBS snapshots. After they are stored as snapshots, whole volumes can be restored as EBS volumes and attached to EC2 instances, accelerating a cloud-based DR solution. The volumes can also be restored to Storage Gateway, enabling your on-premises applications to revert back to a previous state.



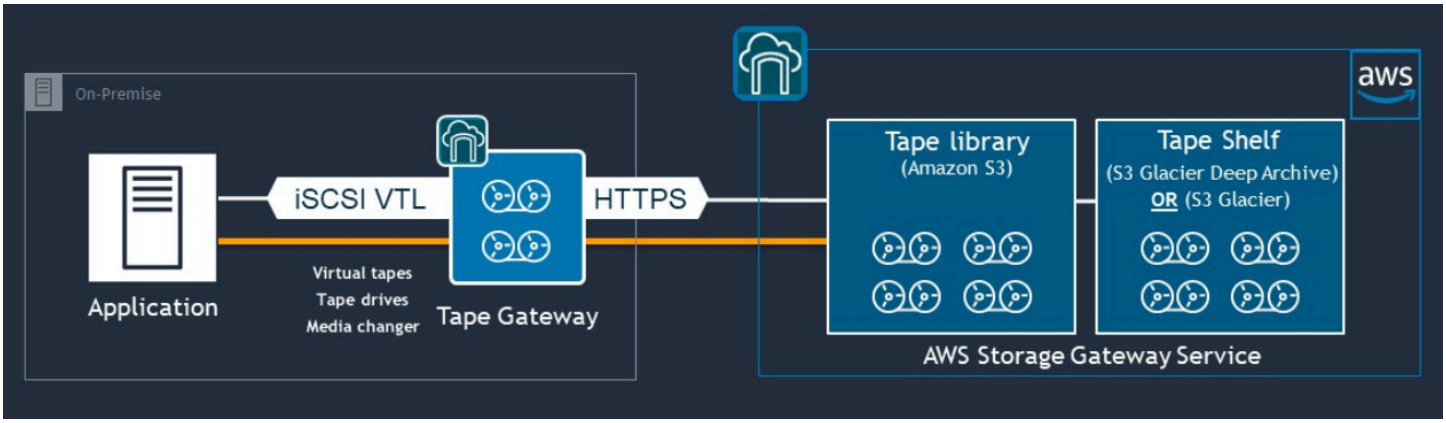
Because a volume gateway integrates with the Amazon EBS volume feature of Amazon EC2, you can use AWS Backup to automate and schedule your snapshot process. A volume gateway provides you with the added benefits of durable, Amazon S3–backed Amazon EBS snapshots and tagging features. For more information, see the Amazon EBS snapshot documentation.

Tape gateway

A tape gateway offers the high durability, low-cost tiered storage, and extensive features of Amazon S3 for your offsite virtual tape backup store. All your virtual tapes stored in Amazon S3 are replicated and stored across at least three geographically dispersed Availability Zones. Your virtual tapes are protected by 11 nines of durability.

AWS also performs fixity checks on a regular basis to confirm that your data can be read and that no errors have been introduced. All tapes stored in Amazon S3 are protected by server-side encryption using default keys or your AWS KMS keys. In addition, you avoid physical security risk associated with tape portability. With a tape gateway, you get correct data, compared to offsite warehousing of tapes, where you might receive an incorrect or broken tape during restore.

You can save on monthly storage costs when storing your data in Amazon S3. You can save even more for your long-term archival requirements by using S3 Glacier Deep Archive.



A tape gateway acts as a virtual tape library (VTL) that spans from your on-premises environment to highly scalable, redundant, and durable storage services: Amazon S3, S3 Glacier Flexible Retrieval, and S3 Glacier Deep Archive.

The tape gateway presents Storage Gateway to your existing backup application as an open standard iSCSI-based VTL, with a virtual media changer and virtual tape drives. You can continue to use your existing backup applications and workflows while writing to a collection of virtual tapes stored on massively scalable Amazon S3. When you no longer require immediate or frequent access to the data on a virtual tape, your backup application can archive it into S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive, further reducing storage costs.

You can retrieve a tape that is archived in S3 Glacier Flexible Retrieval or S3 Glacier Deep Archive typically in 3–5 hours or 12 hours, respectively. The tape gateway can be used with a backup application that is compatible with the iSCSI-based tape library interface for accessing the virtual tapes. Also consider the minimum 100-GB storage size per tape. For more information, review the list of third-party backup applications that support tape gateways.

Backup and recovery of applications from AWS to your data center

You might have a policy requiring you to implement a scenario such as DR or business continuity for your cloud-based workloads and your on-premises infrastructure. If you already have a data backup framework for your on-premises servers, you can extend it to your AWS resources over a VPN connection or through AWS Direct Connect. You can install the backup agent on the EC2 instances and back up your data and applications according to your data-protection policies. You can also use Amazon S3 as the intermediate service to store your application-level backups. You can then use the API operations, SDKs, or the AWS CLI to restore the data to your on-premises environment.

To back up data in AWS services other than Amazon EC2, use the AWS CLI, SDKs, and API operations to extract the data into your desired format. Then copy the data to Amazon S3, and copy it from Amazon S3 to your on-premises environment. Some services provide direct export to Amazon S3. For example, Amazon RDS supports native backup of Microsoft SQL Server databases to Amazon S3.

Backup and recovery of cloud-native AWS services

Your backup and recovery approach should cover the AWS services that are used in your workloads. AWS provides service-specific features and options for managing and interacting with your data. You can use the console, the AWS CLI, SDKs, and API operations to implement backup and recovery for the AWS services that you are using. This guide covers Amazon RDS and Amazon DynamoDB as examples. AWS Backup supports both DynamoDB and Amazon RDS and should be used if it satisfies your requirements.

Backup and recovery for Amazon RDS

Amazon RDS includes features for automating database backups. Amazon RDS creates a storage volume snapshot of your database instance, backing up the entire DB instance, not individual databases only. Using Amazon RDS, you can establish a backup window for automated backups, create database instance snapshots, and share and copy snapshots across Regions and accounts.

Amazon RDS provides two different options for backing up and restoring your DB instances:

- **Automated backups** provide point-in-time recovery (PITR) of your DB instance. Automated backups are turned on by default when you create a new DB instance.

  Amazon RDS performs a daily backup of your data during a backup window that you define when you create the DB instance. You can configure a retention period of up to 35 days for the automated backup. Amazon RDS also uploads the transaction logs for DB instances to Amazon S3 every 5 minutes. Amazon RDS uses your daily backups along with your database transaction logs to restore your DB instance. You can restore the instance to any second during your retention period, up to the `LatestRestorableTime` (typically, the last five minutes).

  To find the latest restorable time for your DB instances, use the `DescribeDBInstances` API call. Or look on the **Description** tab for the database on the Amazon RDS console.

  When you initiate a PITR, transaction logs are combined with the most appropriate daily backup to restore your DB instance to the requested time.
- **DB snapshots** are user-initiated backups that you can use to restore your DB instance to a known state as frequently as you like. You can then restore to that state at any time. You can use the Amazon RDS console or the `CreateDBSnapshot` API call to create DB snapshots. These snapshots are kept until you use the console or the `DeleteDBSnapshot` API call to explicitly delete them.

Both of these backup options are supported for Amazon RDS in AWS Backup, which also provides other features. Consider using AWS Backup to set up a standard backup plan for your Amazon RDS databases, and use the user-initiated instance backup options when your backup plans for a particular database are unique.

Amazon RDS prevents direct access to the underlying storage used by the DB instance. This also prevents you from directly exporting the database on an RDS DB instance to its local disk. In some cases, you can use native backup and restore functions using client utilities. For example, you can use the mysqldump command with an Amazon RDS MySQL database to export a database to your local client machine. In some cases, Amazon RDS also provides augmented options for performing a native backup and restore of a database. For example, Amazon RDS provides stored procedures to export and import RDS database backups of SQL Server databases.

Be sure to thoroughly test your database restore process and its impact on database clients as a part of your overall backup and restore approach.

Using DNS CNAME records to reduce client impact during a database recovery

When you restore a database by using PITR or an RDS DB instance snapshot, a new DB instance with a new endpoint is created. In this way, you can create multiple DB instances from a specific DB snapshot or point in time. There are special considerations when you restore an RDS DB instance to replace a live RDS DB instance. For example, you must determine how you will redirect your existing database clients to the new instance with minimal interruption and modification. You also must ensure continuity and consistency in the data within the database by considering the restored data time and the recovery time when the new instance begins receiving writes.

You can create a separate DNS CNAME record that points to your DB instance endpoint and have your clients use this DNS name. Then you can update the CNAME to point to new, restored endpoint without having to update your database clients.

Set the Time to Live (TTL) for your CNAME record to an appropriate value. The TTL that you specify determines how long the record is cached with DNS resolvers before another request is made. It is important to note that some DNS resolvers or applications might not honor the TTL, and they might cache the record for longer than the TTL. For Amazon Route 53, if you specify a longer value (for example, 172800 seconds, or two days), you reduce the number of calls that DNS recursive resolvers must make to Route 53 to get the latest information in this record. This reduces latency and reduces your bill for the Route 53 service. For more information, see How Amazon Route 53 routes traffic for your domain.

Applications and client operating systems might also cache DNS information that you have to flush or restart to initiate a new DNS resolution request and retrieve the updated CNAME record.

When you initiate a database restore and shift traffic to your restored instance, verify that all your clients are writing to your restored instance instead of your prior instance. Your data architecture might support restoring your database, updating DNS to shift traffic to your restored instance, and then remediating any data that may still be written to your prior instance. If this isn't the case, you can stop your existing instance before you update the DNS CNAME record. Then all access is from your newly restored instance. This may temporarily cause connection problems for some of your database clients that you can handle individually. To reduce client impact, you can perform the database restore during a maintenance window.

Write your applications to handle database connection failures gracefully with retries using exponential backoff. This enables your application to recover when a database connection becomes unavailable during a restore without causing your application to unexpectedly crash.

After you have completed your restore process, you can keep your prior instance in a stopped state. Or you can use security group rules to limit traffic to your prior instance until you are satisfied that it is no longer needed. For a gradual decommissioning approach, first limit access to a running database by the security group. You can eventually stop the instance when it is no longer needed. Finally, take a snapshot of the database instance and delete it.

Backup and recovery for DynamoDB

DynamoDB provides PITR, which makes nearly continuous backups of your DynamoDB table data. When enabled, DynamoDB maintains incremental backups of your table for the last 35 days until you explicitly turn it off.

You can also create on-demand backups of your DynamoDB table by using the DynamoDB console, the AWS CLI, or the DynamoDB API. For more information, see Backing up a DynamoDB table. You can schedule periodic or future backups by using AWS Backup, or you can customize and automate your backup approach by using Lambda functions. For more information about using Lambda functions for backup of DynamoDB, see the blog post A serverless solution to schedule your Amazon DynamoDB On-Demand Backup. If you don't want to create scheduling scripts and cleanup jobs, you can use AWS Backup to create backup plans. The backup plans include schedules and retention policies for your DynamoDB tables. AWS Backup creates the backups and deletes prior backups based on your retention schedule. AWS Backup also includes advanced DynamoDB backup options that aren't available in the DynamoDB service, including lower-cost tiered storage, and cross-account and cross-Region copy. For more information, see Advanced DynamoDB backup.

You must manually set up the following on a restored DynamoDB table:

- Automatic scaling policies
- IAM policies
- Amazon CloudWatch metrics and alarms
- Tags
- Stream settings
- TTL settings

You can restore only the entire table data to a new table from a backup. You can write to the restored table only after it becomes active.

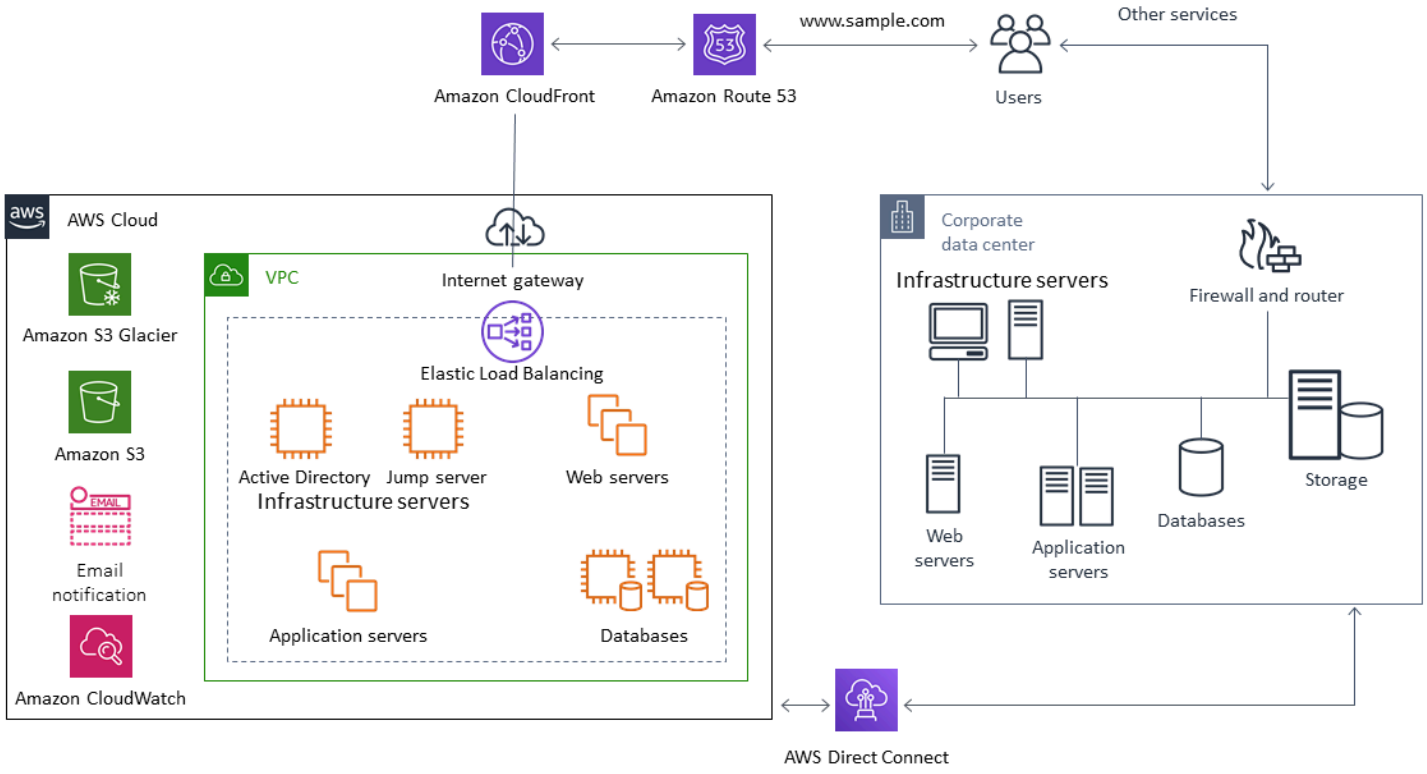
Your restore process must consider how clients will be directed to use the newly restored table name. You can configure your applications and clients to retrieve the DynamoDB table name from a configuration file, AWS Systems Manager Parameter Store value, or another reference that can be updated dynamically to reflect the table name that the client should use.

As a part of the restore process, you should carefully consider your switch-over process. You might choose to deny access to your existing DynamoDB table via IAM permissions and allow access to your new table. You can then update the application and client configuration to use the new table. You might also need to reconcile the differences between your existing DynamoDB table and the newly restored DynamoDB table.

Backup and recovery for hybrid architectures

The cloud-native and on-premises deployments discussed in this guide can be combined into hybrid scenarios where the workload environment has on-premises and AWS infrastructure components. Resources, including web servers, application servers, monitoring servers, databases, and Microsoft Active Directory, are hosted either in the customer data center or on AWS. Applications that are running in the AWS Cloud are connected to applications that are running on premises.

This is becoming a common scenario for enterprise workloads. Many enterprises have data centers of their own and use AWS to augment capacity. These customer data centers are often connected to the AWS network by high-capacity network links. For example, with AWS Direct Connect, you can establish private, dedicated connectivity from your on-premises data center to AWS. This provides the bandwidth and consistent latency to upload data to the cloud for the purposes of data protection. It also provides consistent performance and latency for hybrid workloads. The following diagram provides one example of a hybrid environment approach.



Well-designed data protection solutions typically use a combination of the options described in the cloud-native and on-premises solutions in this guide. Many ISVs provide market leading backup

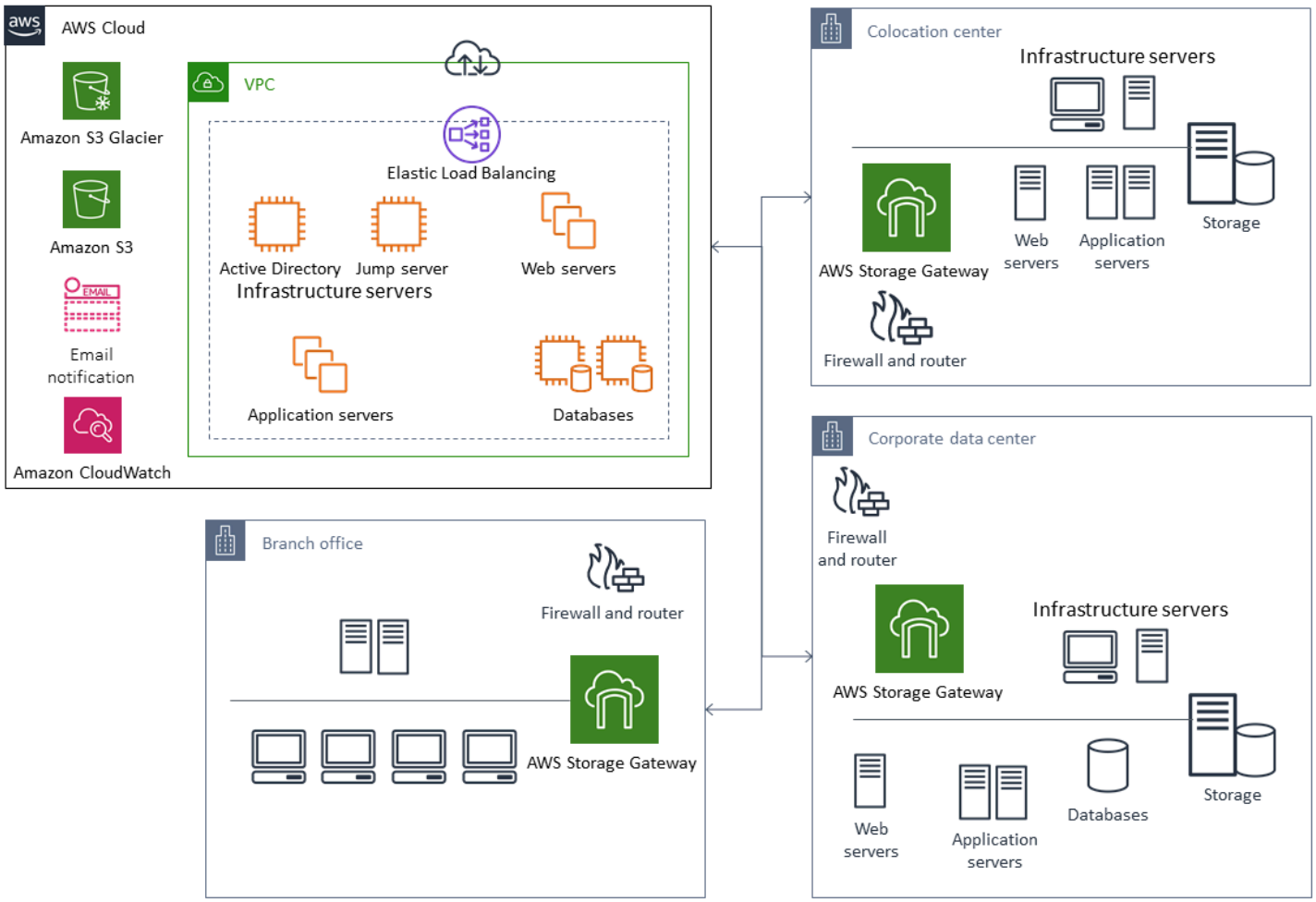
and restore solutions for on-premises infrastructure and have expanded their solutions to support hybrid approaches.

Moving centralized backup management solutions to the cloud for higher availability

By using your existing backup management solution investments with AWS, you can improve the resilience and architecture of your approach. You might have a primary backup server and one or more media or storage servers located on-premises across multiple locations close to the servers and services they are protecting. In this case, consider moving the primary backup server to an EC2 instance to protect it from on-premises disasters and for high availability.

To manage the backup data flows, you can create one or more media servers on EC2 instances in the same Region as the servers they will protect. Media servers near the EC2 instances save you money on internet transfer. When you back up to Amazon S3, media servers increase overall backup and recovery performance.

You can also use Storage Gateway to provide centralized cloud access to data from geographically dispersed data centers and offices. For example, a file gateway gives you on-demand, low-latency access to data stored in AWS for application workflows that can span the globe. You can use features such as cache refresh to refresh data in geographically distributed locations so that content can be easily shared across your offices.


AWS Cloud
Amazon S3 Glacier
Amazon S3
Email notification
Amazon CloudWatch
VPC
Elastic Load Balancing
Active Directory
Jump server
Web servers
Infrastructure servers
Application servers
Databases
Colocation center
Infrastructure servers
AWS Storage Gateway
Web servers
Application servers
Storage
Firewall and router
Corporate data center
Firewall and router
AWS Storage Gateway
Infrastructure servers
Web servers
Application servers
Databases
Storage
Branch office
Firewall and router
AWS Storage Gateway

Disaster recovery with AWS

The backup and restore approaches and supporting services and technologies can be used to implement your disaster recovery (DR) solution. Many enterprises are using the AWS Cloud for backup and restore and as a DR site. AWS provides a number of services and features that support DR and business continuity.

Topics

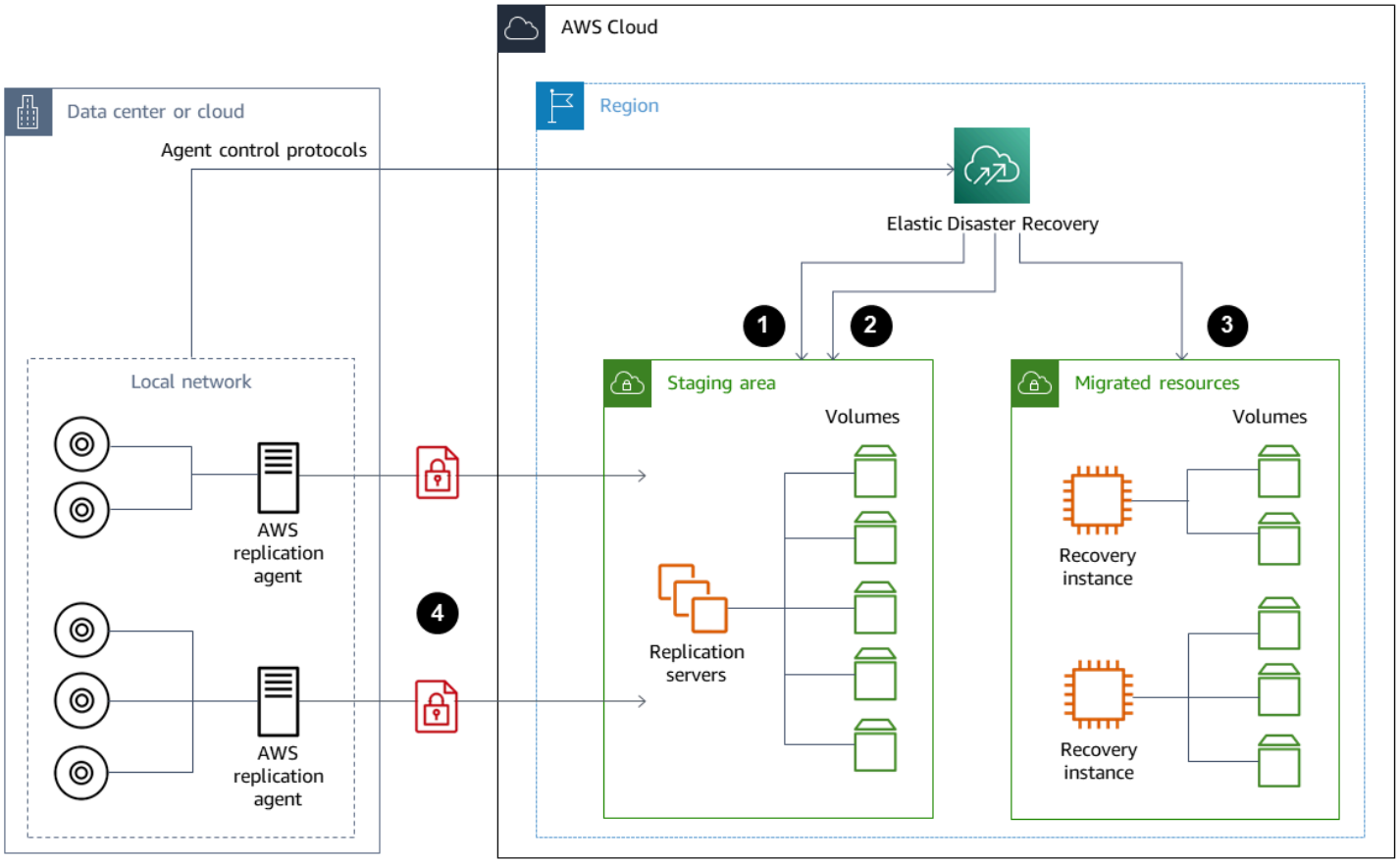
On-premises DR to AWS

Using AWS as an offsite disaster recovery (DR) environment for on-premises workloads is a common hybrid scenario. Define your DR objectives, including the required recovery time and recovery point objectives, before selecting technologies to use. To help with this definition, you can use the DR plan checklist.

There are a number of options available to help you quickly set up and provision a DR environment on AWS. Be sure that you account for all your workload dependencies, and test your DR plan and solution thoroughly and regularly to verify its integrity.

AWS provides AWS Elastic Disaster Recovery for creating a full replica of your on-premises servers, including the root volume and operating system, on AWS. Elastic Disaster Recovery continuously replicates your machines into a low-cost staging area in your target AWS account and preferred AWS Region. The block level replication is an exact replica of your servers' storage including the operating system, system state configuration, databases, applications, and files. If there is a disaster, you can instruct Elastic Disaster Recovery to quickly launch thousands of your machines in their fully provisioned state within minutes.

Elastic Disaster Recovery uses an agent installed on each of your on-premises servers. The agents synchronize the state of your on-premises servers with lower-powered Amazon EC2 equivalents running on AWS. You can also automate your DR failover and failback process with Elastic Disaster Recovery. Automating your failover and failback process can help you achieve a lower and more consistent recovery time objective (RTO).



1. Replication server status reporting
2. Staging area resources automatically created and terminated
3. Recovery instances launched with RTO of minutes and RPO of seconds
4. Continuous block-level replication (compressed and encrypted)

It's important to test the DR process and to verify that the live staging environment doesn't create conflicts with the on-premises environment. For example, confirm that the appropriate licenses are available and functioning in your on-premises, staging, and initiated DR environment. Also confirm that any worker type processes that might poll and pull work from a central database are configured appropriately to avoid overlaps or conflicts. In your DR process, include any necessary steps that must be performed before your recovery server instances come online. Also include the steps to perform after the recovery server instances are online and available. You can use solutions such as the AWS Elastic Disaster Recovery Plan Automation solution or another approach to help you automate your DR plans.

You can use a Storage Gateway volume gateway to provide your on-premises servers with cloud-based volumes. These volumes can also be quickly provisioned for use with Amazon EC2 using Amazon EBS snapshots. In particular, stored volume gateways provide your on-premises applications with low-latency access to their entire datasets. The volume gateways also provide durable snapshot-based backups that can be restored for on premises use or for use with Amazon EC2. You can schedule point-in-time snapshots based on the recovery point objective (RPO) for your workload.

> **Important**
>
> Volume gateway volumes are intended to be used as data volumes and not as boot volumes.

You can use an Amazon EC2 Amazon Machine Image (AMI) with a configuration that matches your on-premises servers and specifies your data volumes separately. After you configure and test the AMI, provision the EC2 instances from the AMI along with the data volumes based on the volume gateway snapshots. This approach requires you to test your environment thoroughly to verify that your EC2 instance is operating properly, especially for Windows workloads.

DR for cloud-native workloads

Consider how your cloud-native workloads align to your DR objectives. AWS provides multiple Availability Zones in Regions around the world. Many enterprises using the AWS Cloud align their workload architectures and DR objectives to withstand the loss of an Availability Zone. The Reliability Pillar in the AWS Well-Architected Framework supports this best practice. You can architect your workloads and their service and application dependencies to use multiple Availability Zones. You can then automate your DR and achieve your DR objectives with minimal to no intervention.

In practice, however, you might find that you are unable to establish a redundant, active, and automated architecture for all of your components. Examine every layer of your architecture to determine the necessary DR processes to achieve your objectives. This might vary from workload to workload, with different architectural and service requirements. This guide covers considerations and options for Amazon EC2. For other AWS services, you can refer to the AWS documentation to determine high availability and DR options.

DR for Amazon EC2 in a single Availability Zone

Try to architect your workloads to actively support and service clients from multiple Availability Zones. You can use Amazon EC2 Auto Scaling and Elastic Load Balancing to achieve a Multi-AZ server architecture for Amazon EC2 and other services.

If your architecture has EC2 instances that can't be load balanced and can have only a single instance running at any given moment, you can use either of the following options.

- Create an Auto Scaling group that has a minimum, maximum, and desired size of 1 and is configured for multiple Availability Zones. Create an AMI that can be used to replace the instance if it fails. Make sure that you define the proper automation and configuration so that a newly provisioned instance from the AMI can be automatically configured and provide service. Create a load balancer that points to the Auto Scaling group and is configured for multiple Availability Zones. Optionally, create an Amazon Route 53 alias that points to the load balancer endpoint.
- Create a Route 53 record for your active instance and have your clients connect using this record. Create a script that creates a new AMI of your active instance and uses the AMI to provision a new EC2 instance in the stopped state in a separate Availability Zone. Configure the script to run periodically and to terminate the previous stopped instance. If there is an Availability Zone failure, start your backup instance in your alternative Availability Zone. Then update the Route 53 record to point to this new instance.

Test your solution thoroughly by simulating the failure that the solution was designed to protect against. Also consider the updates that your DR solution will need as your workload architecture changes.

DR for Amazon EC2 in a regional failure

Customers with very high availability requirements (for example, mission-critical applications that cannot tolerate any downtime) can use AWS across multiple Regions to provide further resiliency against issues at the Region level. Customers must carefully weigh the complexity, cost, and effort required to establish and maintain a multi-Region DR plan against the benefit. AWS provides features that support multi-Region architectures for global availability, failover, and DR. This guide covers a few of the available features that are specific to backup and recovery for Amazon EC2.

AWS AMIs and Amazon EBS snapshots are regional resources that can be used to provision new instances within a single Region. However, you can copy your snapshots and AMIs to another Region and use them to provision new instances in that Region. To support a regional failure

DR plan, you can automate the process of copying AMIs and snapshots to other Regions. AWS Backup and Amazon Data Lifecycle Manager support cross-Region copying as a part of your backup configuration.

AWS Elastic Disaster Recovery can be used to automate and continuously replicate your Amazon EC2 servers in one Region to an alternate DR Region. Elastic Disaster Recovery can simplify your multi-Region DR approach and help you to regularly test your cross-Region Amazon EC2 DR plan by using drills. Elastic Disaster Recovery can help when backup and recovery is unable to meet your RTO and RPO objectives. Elastic Disaster Recovery can help you lower your RTO to minutes and your RPO into the sub-second range.

Whichever solution you use, you must determine the provisioning, failover, and failback process to use in the event of an outage. You can use Route 53 with health checks and Domain Name System failover to help support your solution.

Cleaning up backups

To reduce costs, clean up the backups that are no longer required for recovery or retention purposes. You can use AWS Backup and Amazon Data Lifecycle Manager to automate your retention policy for a portion of your backups. However, even with these tools in place, you still need a cleanup approach for backups that are taken separately.

A tagging strategy is a prerequisite to a cleanup strategy. Use tagging to identify resources that should be cleaned up, notify owners appropriately, and automate your cleanup process. Backups created by AWS have creation dates aligned to them, but tagging is important to correlate backups to your workloads, retention requirements, and restore-point identification.

You can implement a cleanup process for snapshots using automation. For example, you can scan your account for snapshots and determine if the corresponding volumes are in an attached state or an available state. You can further filter the results on a time threshold that you specify. Using the tags attached to the volume, you can automatically send email to snapshot owners, and warning them that their snapshots have been scheduled for deletion. This automated remediation can be implemented by using AWS Config rules, a script using the AWS CLI, or a Lambda function using the AWS SDK.

Systems Manager provides the AWS-DeleteEBSVolumeSnapshots and AWS-DeleteSnapshot documents to help you initiate and automate the cleanup of Amazon EBS snapshots. You can also use the AWS CLI and AWS SDK to automate the cleanup of other AWS resources such as Amazon RDS snapshots.

Backup and recovery FAQ

What backup schedule should I select?

Define a backup schedule frequency that aligns to your recovery point objective (RPO). Define a backup time when your workload is under the least amount of load and when user impact can be reduced. Create a point-in-time snapshot whenever you are going to make a significant change to your workload.

Do I need to create backups in my development accounts?

Test potentially breaking changes in your development accounts for your workloads and create backups before performing breaking changes. You might have many more point-in-time recovery (PITR) backups in your development and non-production accounts from development and testing activities.

Can I upgrade applications and continue to use an EBS volume while a snapshot is being created without any impact?

Snapshots occur asynchronously; the point-in-time snapshot is created immediately, but the status of the snapshot is pending until all the modified blocks have been transferred to Amazon S3. For large initial snapshots or subsequent snapshots where many blocks have changed, the transfer can take several hours. While it is transferring, an in-progress snapshot is not affected by ongoing reads and writes to the volume. For more information, see the AWS documentation.

Next steps

Start by evaluating, implementing, and testing your backup and recovery approach in a non-production environment. It is important to test your recovery process thoroughly and to validate that your restored workloads are operating as expected.

Test the restore process for a single component in your architecture in addition to all components in your architecture. Validate the recovery time for each. Also validate the impact of your backup and restore process on upstream and downstream dependencies. Confirm the impact of any service outage on your upstream dependencies and confirm the downstream impact on your backups.

Additional resources

AWS resources

- AWS Prescriptive Guidance
- AWS documentation
- AWS general reference
- AWS glossary

AWS services

- AWS Backup
- Amazon CloudWatch
- AWS Config
- Amazon DynamoDB
- Amazon EBS
- Amazon EC2
- Amazon EventBridge
- IAM
- Amazon RDS
- Amazon S3
- Storage Gateway
- AWS Systems Manager

Other resources

- Backup and Recovery with AWS Backup (solution)
- Disaster Recovery of Workloads on AWS: Recovery in the Cloud (whitepaper)
- Disaster Recovery Series (AWS Architecture blog posts)
- IT Disaster Recovery Plan Checklist
- Backup and Recovery Approaches Using AWS (technical paper – archived)
- Getting started with AWS Backup

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
Updated information	Updated guidance in the Amazon S3 section.	June 28, 2024
Updated information	Updated information in the On-premises DR to AWS section.	April 13, 2023
Added a section	Added guidance and steps for creating or restoring an instance from a snapshot.	March 7, 2023
Added information about Elastic Disaster Recovery and added clarification	In the Disaster recovery with AWS and Choosing AWS services for data protection sections, added information about AWS Elastic Disaster Recovery. In the Amazon EC2 backup and recovery with snapshots and AMIs, Preparing an EBS volume before creating a snapshot or AMI, and Restoring from an Amazon EBS snapshot or an AMI sections, added clarification. Added to the Backup and recovery FAQ.	January 19, 2023
Added a link	Added a link to the Amazon Data Lifecycle Manager documentation in the	October 31, 2022

	Amazon Data Lifecycle Manager section.	
Updated information	Updated the information about restoring volumes.	August 30, 2022
Updated information and added new section	In the Choosing AWS services for data protection section, added services. Added the section Backup and recovery using AWS Backup. In the *Backup and recovery using Amazon S3 and Amazon S3 Glacier* section, added information about new Amazon S3 Glacier storage classes. In the Backup and recovery for Amazon EC2 with EBS volumes section, added links to documentation and additional information. In the Backup and recovery of cloud-native AWS services section, added a recommendation to use AWS Backup. In the Additional resources section, added resources.	January 28, 2022
Updated information	Added information about setting storage classes to the *S3 Glacier Flexible Retrieval* section. Added information about retrieving snapshots to the Amazon EC2 backup and recovery with snapshots and AMIs section.	September 9, 2021

Updated information	In the AWS Backup section, added information about the AWS services that AWS Backup supports.	June 1, 2021
Initial publication	—	July 29, 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, 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.