**Amazon S3**

**Storage services**

* **Storage Services**: Overview of AWS storage services and their functionalities. Select storage services by keeping in mind size, performance and cost of your use case.
* **S3 Storage Class**: Different classes of S3 storage, such as Standard, Infrequent Access, and Glacier.
* **S3 Terminology**: Key terms like buckets, objects, and keys.
* **S3 Advanced Features**: Advanced functionalities like versioning, lifecycle policies, and cross-region replication.
* **Creating S3 Buckets Lab**: Practical steps to create and configure S3 buckets.
* **S3 Bucket Properties**: Properties and settings of S3 buckets, including permissions and logging.
* **S3 Managing Objects Lab**: Practical steps to manage objects within S3 buckets, including uploading, downloading, and deleting.

Additional AWS Storage Services

* **Glacier**: Overview of Glacier storage for long-term archival.
* **Setting up a Glacier Vault Lab**: Practical steps to set up and manage Glacier vaults.
* **S3 and Tape Gateway**: Integration of S3 with Tape Gateway for hybrid storage solutions.
* **S3 Enhanced Features**: Additional features like Transfer Acceleration and Event Notifications.
* **Elastic Block Store (EBS)**: Overview of EBS for block storage.
* **Creating EBS Volumes Lab**: Practical steps to create and manage EBS volumes.
* **Elastic File System (EFS)**: Overview of EFS for scalable file storage.
* **Creating an EFS File System Lab**: Practical steps to create and manage EFS file systems.
* **EFS and PrivateLink**: Integration of EFS with PrivateLink for secure access.
* **Intro to Amazon FSx**: Introduction to Amazon FSx for Windows File Server and Lustre.
* **Hands on with FSx**: Practical steps to use FSx.
* **Integrating On-Premises Storage**: Integration of on-premises storage with AWS.
* **Storage Access Security Lab**: Practical steps to secure storage access.
* **Storage Performance**: Techniques to optimize storage performance.

**S3 storage class**

* **Object Storage**: S3 is designed for storing objects, which can be any chunk of data, such as files. These objects are stored in "buckets".
* **Storage Classes**: Different S3 storage classes offer varying levels of availability and cost:
  + **Standard**: High availability and durability, with data distributed across at least three Availability Zones.
  + **1A Class**: Lower availability and cost, using only one Availability Zone.
* **Security Features**: S3 supports encryption and automatic data classification for security and management purposes.
* **Data Classification**: Helps in managing data by classifying it based on usage, enabling automatic processing like moving infrequently accessed data to Glacier.
* **Big Data Analytics**: Allows performing analytics directly on data stored in S3 buckets without needing to move it to a database.
* **Data Ingestion Methods**: Various ways to get data into S3, including:
  + **AWS APIs**: For programmatic access.
  + **Amazon Direct Connect**: VPN connection for data transfer.
  + **Storage Gateway**: Local storage replication to S3.
  + **Kinesis Firehose**: For large-scale data ingestion.
  + **Transfer Acceleration**: Optimized data transfer using CloudFront.
  + **Snow Family**: Physical devices (Snowball, Snowball Edge, Snowmobile) for transferring large amounts of data.

**S3 terminology**

* **Buckets**: Containers in S3 where you store your data (objects). You can have multiple buckets, and they are essential for organizing your data.
* **Regions**: When you create a bucket, you assign it to a specific region. This is important for data accessibility and latency.
* **Objects**: The actual data stored in S3 buckets, similar to files on your local computer.
* **Keys**: Unique identifiers for objects within a bucket, similar to file names but more flexible.
* **Prefixes**: Used to create a logical hierarchy within buckets, making it appear as if there are folders, even though S3 doesn't have a traditional folder structure.
* **Object URLs**: Each object in S3 has a URL, which can be used to access the object over the internet.
* **Eventual Consistency**: Changes to objects in S3 might not be immediately visible across all regions. It takes some time for updates to propagate, ensuring eventual consistency.
* **Static Website Hosting**: S3 can host static websites, serving HTML files directly from a bucket.
* **Common Operations**: Includes creating/deleting buckets, writing/reading/deleting objects, and managing object properties.
* **REST API**: The API used to interact with S3, mapping HTTP methods (like GET, PUT, POST, DELETE) to CRUD operations (Create, Read, Update, Delete).

These concepts are fundamental to understanding and working with Amazon S3.

**S3 advanced features**

* **Prefixes and Delimiters**: In S3, there are no traditional folders. Instead, you use prefixes (like "marketing/plans/") to organize your data, making it appear as if there are folders.
* **Storage Classes**: Different ways to store data in S3:
  + **S3 Standard**: High availability and performance, but more expensive.
  + **S3 Infrequent Access (IA)**: Cheaper, for data accessed less frequently.
  + **S3 Reduced Redundancy Storage (RRS)**: Least expensive, less redundancy.
  + **Glacier**: Very cheap for storage but expensive for frequent retrievals.
* **Object Lifecycle Management**: Automates moving data between storage classes. For example, a file can start in S3 Standard, move to S3 IA after 60 days, and then to Glacier after another 60 days.
* **Encryption**:
  + **Server-Side Encryption**: AWS encrypts your data after it’s uploaded.
  + **Client-Side Encryption**: You encrypt your data before uploading it to AWS.
* **Versioning**: Keeps multiple versions of an object. Once enabled, it cannot be disabled, only suspended.
* **Additional Features**:
  + **MFA Delete**: Requires multi-factor authentication to delete objects.
  + **Multipart Upload**: Uploads large files in parts for faster upload.
  + **Range GETs**: Retrieve specific parts of a large file.
  + **Cross-Region Replication**: Replicates data across different regions.
  + **Logging and Event Notifications**: Logs actions on the bucket and notifies you of specific events.

These concepts help in organizing, securing, and managing your data efficiently in S3.

**Creating S3 buckets lab**

* **AWS Management Console**: You can manage S3 buckets through a graphical user interface (GUI) instead of the command line.
* **Creating a Bucket**:
  + **Bucket Name**: Must be DNS compliant and globally unique across all of Amazon S3.
  + **Region**: Choose a region for your bucket, such as US East Ohio.
  + **Properties**: Configure properties like versioning, logging, tags, and encryption.
* **Tags**: Tags help in organizing and searching through your buckets. You can add multiple tags like "department" and "purpose".
* **Permissions**: Set permissions for who can access the bucket, including public access and system permissions for logging.
* **Review and Create**: Review your settings and create the bucket.
* **Bucket Properties**: After creation, you can configure additional properties like versioning, server access logs, and encryption.
* **Versioning**: Once enabled, versioning cannot be disabled, only suspended. It keeps multiple versions of objects in the bucket.

These steps and concepts help you efficiently create and manage S3 buckets in AWS.

**S3 bucket properties**

* **Static Web Hosting**: You can use an S3 bucket to host a static website by uploading HTML, image, and audio files. The bucket will have a URL for accessing the website, but it's recommended to use a DNS redirector for a user-friendly URL.
* **Encryption**:
  + **AES 256**: AWS manages the encryption keys for you, providing a simple and automatic encryption solution.
  + **KMS (Key Management Service)**: You manage the encryption keys, giving you more control and flexibility, but requiring additional management.
* **Permissions**:
  + **Bucket-Level Permissions**: These permissions apply to all objects in the bucket and can be inherited by individual objects.
  + **Public Access**: You can grant public access to the bucket, allowing anyone to list, write, or read objects.
  + **Bucket Policy**: Use JSON policies or a visual editor to set permissions.
  + **CORS (Cross-Origin Resource Sharing)**: Allows web applications to interact with the bucket.
* **Management**:
  + **Lifecycle Rules**: Set rules for transitioning objects to different storage classes (e.g., infrequent access, Glacier) or for automatic expiration (deletion) of objects after a certain period.
  + **Replication**: Enable cross-region replication for disaster recovery.
  + **Analytics, Metrics, and Inventory**: Monitor and manage the bucket's performance and usage.

These concepts help you manage and secure your S3 buckets effectively.

Top of Form

**S3 managing objects lab**

* **Uploading Files**: You can upload files to your S3 bucket. AWS S3 allows you to create a structure that looks like folders, but they are actually just prefixes and delimiters.
* **Permissions**: When uploading files, you can set permissions to control who can access them. You can add other AWS accounts or make files public, but be cautious with public access.
* **Storage Classes**: You can choose different storage classes (like Standard, Standard-IA) based on how frequently you need to access the files. Each class has different costs and retrieval fees.
* **Versioning**: S3 allows you to keep multiple versions of a file. If you upload a new version of a file, the old version is still accessible. This is useful for tracking changes or recovering previous versions.
* **Encryption**: You can encrypt your files for security. AWS offers server-side encryption options like AES-256.
* **Tags and Metadata**: You can add tags and metadata to your files for better management and lifecycle policies. Tags can help in organizing and applying specific rules to files.
* **Lifecycle Management**: You can set rules to automatically transition files to different storage classes or delete them after a certain period.

These concepts help you manage your data efficiently in AWS S3, ensuring security, cost-effectiveness, and ease of access.

**Additional AWS Storage Services**

**Glacier**

* **Glacier Overview**: Glacier is an archival storage solution in AWS designed for data that you don't need to access frequently. It's cost-effective, with storage costs being fractions of a penny per gigabyte per month.
* **Access Methods**: There are three ways to retrieve data from Glacier:
  + **Expedited**: Fastest retrieval (3-5 minutes) but most expensive.
  + **Standard**: Takes 3-5 hours and is less expensive.
  + **Bulk**: Takes 5-12 hours and is the least expensive.
* **Storage Structure**: Data in Glacier is stored in "archives" within "vaults". Vaults are like containers for your archives, and you can secure them with "vault locks" to control access and costs.
* **Integration with S3**: Glacier can be used as a storage class within Amazon S3, allowing you to move infrequently accessed data to Glacier automatically using lifecycle rules.
* **Data Import**: You can use Snow devices to import large amounts of data into Glacier, which is useful for offsite storage and disaster recovery.
* **Cost Management**: Up to 5% of your data can be retrieved for free each month, but there's no rollover. You can also configure vaults to limit retrieval costs and control access.

Top of Form

Bottom of Form

**Setting up a Glacier vault lab**

* **Creating a Glacier Vault**: To create a Glacier vault, start by accessing the Glacier Management Console through the AWS Services Interface. Select "Glacier" under the Storage section and click on "Create Vault".
* **Choosing a Region**: Decide where your vault will be located. Typically, you would choose the region where your other AWS resources are located.
* **Naming the Vault**: Give your vault a name that helps you identify it easily.
* **Event Notifications**: You can enable notifications to get alerts when actions are taken in the vault. You can create a new SNS topic for notifications or use an existing one.
* **Vault Properties**: Once the vault is created, you can view its properties, such as the region, creation date, and Amazon Resource Name (ARN). You can also configure retrieval policies to manage costs.
* **Storing Data**: After setting up the vault, you can store data in it through various methods like Storage Gateway, application APIs, or the command line interface. Data stored in Glacier is referred to as "archives".

**S3 and Tape Gateway**

* **Tape Gateway**: AWS Tape Gateway allows you to use cloud storage for tape backups. It creates a virtual tape library (VTL) that emulates physical tape drives.
* **Integration with S3**: The Tape Gateway works with Amazon S3, where the virtual tapes are stored as objects in S3 buckets.
* **Glacier Deep Archive**: You can configure the virtual tapes to be stored in Glacier Deep Archive, which is a very low-cost storage option for data that is rarely accessed.
* **Setup**: To set up the Tape Gateway, you need to go to the Storage Gateway section in the AWS Management Console. You can choose to host the Tape Gateway on an EC2 instance or within your VPC (Virtual Private Cloud).
* **Usage Scenarios**: The Tape Gateway can be used by EC2 instances or on-premises systems that need to back up data to tape drives. Depending on your needs, you can make the virtual tape library publicly accessible or restrict it to your VPC.
* **Cost Efficiency**: Storing data in Glacier Deep Archive is very cost-effective, but it is designed for infrequent access, similar to traditional tape backups.

**S3 enhanced features**

* **Intelligent Tiering**: This feature allows you to automatically move data to the most cost-effective storage tier based on usage patterns. You set lifecycle rules to transition objects to different tiers, like moving data to Glacier after 90 days of infrequent access to save costs.
* **Object Locking**: This feature enables you to lock objects in S3 so they cannot be modified or deleted. It's useful for compliance and data protection. You can only enable object locking when creating a bucket, and it requires bucket versioning to be enabled.
* **Batch Processing**: This feature allows you to perform bulk operations on S3 objects, such as moving, deleting, or changing storage classes for multiple objects at once. You create a job based on a manifest (a list of objects) to automate these tasks.

**Elastic Block Store (EBS)**

* **Elastic Block Store (EBS)**: EBS provides block-level storage for use with Amazon EC2 instances. It's like a hard drive in the cloud, providing persistent storage that remains even if the instance is stopped or terminated.
* **Types of EBS Volumes**:
  + **Magnetic Volumes**: These are the lowest cost and slowest option, suitable for infrequently accessed data.
  + **SSD (Solid State Drive) Volumes**: These offer faster performance. There are two types:
    - **General Purpose SSD**: Suitable for most workloads.
    - **Provisioned IOPS SSD**: Provides high performance with guaranteed IOPS (Input/Output Operations Per Second) for demanding applications.
* **Snapshots**: Snapshots are backups of EBS volumes. They can be used to restore data or create new volumes with the same data. This is similar to creating an image of a hard drive.
* **Volume Encryption**: EBS volumes can be encrypted to protect data. This ensures that data is secure, even if someone gains physical access to the storage hardware.
* **Volume Recovery**: EBS volumes can be detached from one instance and attached to another, allowing for data recovery and migration.

These concepts help ensure that your data is securely stored and easily recoverable, providing flexibility and reliability for your cloud applications.

**Creating EBS volumes lab**

* **Finding EBS Volumes**: Unlike other AWS storage solutions, EBS volumes are found under the EC2 dashboard, not in the storage section.
* **Creating a Volume**: To create an EBS volume, you go to the EC2 dashboard, select "Elastic Block Store," and then "Volumes." Click on "Create Volume" to start the process.
* **Volume Types**: You can choose from different volume types:
  + **General Purpose SSD**: Balanced performance.
  + **Provisioned IOPS SSD**: High performance with guaranteed IOPS.
  + **Magnetic Standard**: Basic, slower, and available in the free tier.
  + **Cold HDD**: Large capacity but slow.
  + **Throughput Optimized HDD**: Faster than Cold HDD.
* **Volume Size**: You can specify the size of the volume, from 1 GB to 1 TB.
* **Availability Zone**: Ensure the EBS volume is in the same availability zone as your EC2 instance for optimal performance.
* **Snapshots**: You can create a volume from a snapshot, which is a backup of another volume.
* **Encryption**: You can choose to encrypt the volume for added security.
* **Managing Volumes**: Once created, you can manage the volume by attaching it to an instance, creating snapshots, or modifying its properties.

These steps help you set up and manage EBS volumes effectively in AWS.

Top of Form

**Elastic File System (EFS)**

* **Elastic File System (EFS)**: EFS is a scalable file storage service for use with AWS Cloud services and on-premises resources. It allows multiple EC2 instances to access the same file system simultaneously.
* **Shareable Nature**: Unlike EBS, which is attached to a single instance, EFS can be accessed by multiple instances at the same time, making it ideal for shared storage.
* **Hierarchical Structure**: EFS has a hierarchical file system structure, similar to traditional file systems, with directories and subdirectories.
* **NFS Protocol**: EFS uses the Network File System (NFS) protocol, specifically NFS version 4, for accessing the file system.
* **Linux Compatibility**: EFS is supported on Linux instances, but not on Windows instances. For Windows, you can use EBS volumes and share folders from the Windows instance itself.
* **Performance**: EFS offers low and consistent latency. It is suitable for various use cases like web serving, content management, enterprise applications, and big data analytics.
* **Data Availability and Durability**: EFS stores data redundantly across multiple availability zones, ensuring high availability and durability.

These concepts highlight the flexibility and scalability of EFS for shared storage solutions in AWS.

**Creating an EFS file system lab**

* **EFS File System**: EFS stands for Elastic File System, which is a scalable file storage service in AWS. The term "EFS file system" is used redundantly but refers to an instance of EFS.
* **Creating an EFS**:
  + Navigate to the AWS services listing and select EFS under the storage section.
  + Click on "Create file system" and choose the Virtual Private Cloud (VPC) it will be associated with. The default VPC can be used if no specific VPC is needed.
* **Mount Targets**: EFS creates mount targets in multiple availability zones (e.g., US East 2A, 2B, 2C) to ensure that instances in different zones can access the file system efficiently.
* **Configuration Options**:
  + **Subnets**: You can select specific subnets within your VPC.
  + **Elastic IP**: Assign an elastic IP address manually or let it be automatic.
  + **Security Group**: Set firewall rules for accessing the EFS.
* **Performance and Throughput Modes**:
  + **General Purpose**: Suitable for most applications.
  + **Max IO**: Ideal for applications with high throughput needs.
  + **Bursting**: Escalates throughput as needed.
  + **Provisioned**: Guarantees a specific throughput level.
* **Encryption**: You can enable encryption for added security.
* **Final Steps**: After configuring the settings, review and create the file system. Once created, you can see the file system and its mount targets.
* **Mounting Instructions**:
  + EFS can be mounted using NFS (Network File System) on Linux systems.
  + Windows does not support direct EFS connections, but workarounds like using a Linux proxy are possible.

These steps and concepts will help you understand how to create and configure an EFS file system in AWS.

**EFS and PrivateLink**

* **Elastic File System (EFS)**: EFS is used to create network-attached storage solutions in the cloud. It allows multiple AWS accounts to access and store data in a central EFS location.
* **PrivateLink**: PrivateLink is a technology that enables private connectivity between VPCs and AWS services. It uses Elastic Network Interfaces (ENIs) with fixed IP addresses for secure and private access.
* **Creating Endpoints**:
  + Navigate to the VPC section in AWS.
  + Select "Endpoints" and click on "Create Endpoint".
  + Choose the AWS service you want to connect to, such as EFS.
  + An interface endpoint powered by PrivateLink will be created, using an ENI as the entry point.
* **Cost Considerations**: Each endpoint created with PrivateLink has a fixed IP address, which incurs a monthly cost. Plan endpoints carefully to manage costs effectively.

These concepts explain how EFS and PrivateLink work together to provide secure, scalable, and cost-effective storage solutions in AWS.

**Intro to Amazon FSx**

* **Amazon FSx**: A cloud-based file system that allows you to create and manage different types of file systems without needing to set up and maintain physical servers.
* **Types of File Systems**:
  + **Windows File Systems**: Allows you to create Windows server shares without having a physical Windows server.
  + **Lustre File Systems**: A high-performance file system suitable for intensive processing tasks, like storing and processing graphics files.
* **Benefits**:
  + **Simplicity and Management**: FSx handles all the preparatory tasks, such as building the server and integrating with Active Directory, so you can focus on creating and managing folders and shares.
  + **AWS Integration**: FSx integrates with other AWS services, such as CloudWatch, for log storage and versioning capabilities.
  + **Native Compatibility**: Supports applications that require server message block (SMB) shares, ensuring compatibility with existing systems.
  + **Cost Optimization**: Reduces costs by eliminating the need for full-blown servers to manage file shares, using less capable systems for file sharing.

These points highlight the main features and advantages of using Amazon FSx for your cloud storage needs.

Top of Form

**Hands on with FSx**

* **FSx File System Setup**: The video demonstrates the process of setting up an FSx file system in AWS.
* **Types of FSx File Systems**:
  + **FSx for Windows File Server**: Chosen in this demonstration.
  + **FSx for Lustre**: Another option available.
* **Configuration Steps**:

1. **File System Name**: Optional but recommended for easier management.
2. **Deployment Type**: Choose between Single-AZ (single availability zone) or Multi-AZ (multiple availability zones). Single-AZ is used in this example.
3. **Storage Capacity**: Specify the size in gibibytes.
4. **Throughput Capacity**: Can be set to recommended or specified in megabytes per second.
5. **Network Security**: Set the VPC and security groups. Defaults can be used.
6. **Windows Authentication**: Choose between AWS-managed Active Directory or self-managed Active Directory.
7. **Encryption**: Set up encryption options.
8. **Tags**: Optional tags for administration and management.

* **Final Steps**: After configuring the settings, click "Next" to complete the setup and create the FSx file system.

These steps and concepts will help you understand how to set up and configure an FSx file system in AWS.

**Integrating on-premises storage**

* **AWS Storage Gateway**: A software appliance that helps integrate on-premises storage with AWS cloud storage. It runs as a virtual machine on VMware ESX or Hyper-V.
* **Types of Storage Gateways**:
  + **File Gateway**: Uses the Network File System (NFS) to store files in Amazon S3 buckets. Supports Windows systems.
  + **Volume Gateway**: Uses the iSCSI protocol for block-level storage, turning the AWS cloud into a storage area network (SAN).
  + **Tape Gateway**: Provides a virtual tape library for backup software, eliminating the need for physical tape rotation by storing backups directly in the AWS cloud.
* **Deployment Options**:
  + **On-Premises**: The gateway runs as a virtual machine in your local network.
  + **Amazon EC2 Instance**: Creates a storage gateway between virtual private clouds (VPCs) within AWS.
* **Planning Deployment**: Choose the appropriate storage solution (file, volume, or tape), consider performance expectations, and decide on the hosting option (on-premises or EC2 instance).

These concepts explain how to integrate local storage with AWS using the Storage Gateway, providing flexibility and scalability for your storage needs.

**Storage access security lab**

* **S3 Bucket Permissions**: You can set permissions on S3 buckets to control who can access and manage the data stored in them. This is crucial for ensuring that only authorized individuals can access sensitive or important data.
* **Identity and Access Management (IAM)**: IAM is used to manage who can do what within the AWS management consoles. It plays a significant role in determining management and administrative permissions.
* **Management Console and CLI**: Permissions can be managed through the AWS Management Console or using command-line interface (CLI) commands. For the exam, it's important to know the high-level commands like S3 API.
* **Canonical ID and ARN**: When adding users for permissions, you can use a canonical ID or an email address. However, using the Amazon Resource Name (ARN) is often more practical. The ARN uniquely identifies AWS resources and can be used in JSON-based policies.
* **Bucket Policy and Policy Generator**: Instead of manually writing JSON policies, you can use the policy generator to create policies that define permissions. This tool simplifies the process by allowing you to specify actions (allow or deny) and the resources they apply to.
* **Policy Elements**: Policies can include multiple statements with different actions and principles. The policy generator helps you create these policies by providing a user-friendly interface.

These points highlight the main features and steps involved in managing storage access security for S3 buckets in AWS.

**Storage performance**

* **IOPS (Input/Output Operations Per Second)**: This is a performance measurement for storage devices. For example, general purpose SSDs (GP2) can provide up to 10,000 IOPS, while provisioned IOPS SSDs can go up to 32,000 IOPS. Magnetic hard drives have much lower IOPS, ranging from 250 to 500.
* **SSD vs. Magnetic Hard Drives**: SSDs (Solid State Drives) offer significantly higher performance compared to magnetic hard drives. General purpose SSDs are the default choice for new EBS volumes due to their balance of performance and cost. Magnetic hard drives are more cost-effective but provide lower performance.
* **Storage Sizes**: AWS uses terms like GiB (Gibibytes) and TiB (Tebibytes) instead of GB (Gigabytes) and TB (Terabytes). For example, general purpose SSDs can range from 1 GiB to 16 TiB in size.
* **S3 Storage Classes**: Different S3 storage classes offer varying levels of durability and availability:
  + **S3 Standard**: 99.999999999% (11 nines) durability and 99.99% (4 nines) availability.
  + **S3 Standard-IA (Infrequent Access)**: Same durability but 99.9% (3 nines) availability.
  + **S3 One Zone-IA**: Same durability but 99.5% availability.

These concepts highlight the importance of choosing the right storage type and class based on performance needs and cost considerations.

Top of Form

Bottom of Form

Bottom of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form

Bottom of Form

Top of Form

Bottom of Form

Top of Form

Bottom of Form