**Storage/Caching options available in AWS cloud**

|  |  |
| --- | --- |
| Services | Uses |
| S3 | Scalable storage in cloud |
| Glacier | Low-cost archive storage in the cloud |
|  |  |
| EBS | Persistent block storage volumes for Amazon EC2 virtual machines |
|  |  |
| EC2 Instance Storage | Temporary block storage volumes for Amazon EC2 virtual machines |
|  |  |
| Import/Export | Large volume data transfer |
|  |  |
| Storage Gateway | Integrates on-premises IT environments with cloud storage |
|  |  |
| CloudFront | Global content delivery network (CDN) |
|  |  |
| SQS | Message queue service |
|  |  |
| RDS | Managed relational database server for MySQL, Oracle, and Microsoft SQL Server |
|  |  |
| DynamoDB | Fast, predictable, highly-scalable NoSQL data store |
|  |  |
| Elastic Cache | In-memory caching service |
|  |  |
| Red shift | Fast, powerful, full-managed, petabyte-scale data warehouse service |
|  |  |
| Databases on Amazon | Self-managed database on an Amazon EC2 instance |
| EC2 |  |

**Amazon Simple Storage Service**(S3)

Usage patterns:

1. One very common use for Amazon S3 is storage and distribution of static web content and media. This content can be delivered directly from Amazon S3, since each object in Amazon S3 has a unique HTTP URL address, or Amazon S3 can serve as an origin store for a content delivery network (CDN), such as Amazon Cloud Front.

2. Amazon S3 is also frequently used to host entire static websites. Amazon S3 provides a highly available and highly scalable solution for websites with only static content, including HTML files, images, videos, and client-side scripts such as JavaScript

3. Amazon S3 is also commonly used as a data store for computation and large-scale analytics, such as analyzing financial transactions, clickstream analytics, and media transcoding. Because of the horizontal scalability of Amazon S3, you can access your data from multiple computing nodes concurrently without being constrained by a single connection.

4. Amazon S3’s versioning capability is available to protect critical data from inadvertent deletion.

Cost Model:

Amazon S3 has three pricing components: storage (per GB per month), data transfer in or out (per GB per month), and requests (per n thousand requests per month)

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Services |
| File system | Amazon S3 uses a flat namespace and isn’t meant to serve as a standalone, POSIX-compliant file system. Instead, consider using Amazon Elastic File System (Amazon EFS) as a file system. | Amazon Elastic File System |
| Structured data with query | Amazon S3 doesn’t offer query capabilities to retrieve specific objects. When you use Amazon S3, we expect that you know the exact bucket name and key for the files you want to retrieve from the service. Thus, Amazon S3 cannot be used as a database or search engine by itself. Instead, you can pair Amazon S3 with Amazon DynamoDB, Amazon CloudSearch or Amazon Relational Data Service (Amazon RDS) to index and query metadata about Amazon S3 buckets and objects | Amazon Dynamo DB  Amazon Cloud Search  Amazon RDS |
| Rapidly changing data | Data that must be updated very frequently might be better served by storage solutions with lower read/write latencies such as Amazon EBS volumes, Amazon RDS, Amazon DynamoDB, or relational databases running on Amazon EC2 | Amazon EBS volumes  Amazon RDS  Amazon Dynamo DB  Amazon EC2 |
| Archival data | Data that requires encrypted archival storage with infrequent read access with a long recovery time objective (RTO) can be stored in Amazon Glacier more cost-effectively. | Amazon Glacier |
| Dynamic website hosting | Although Amazon S3 is ideal for static content websites, dynamic websites that depend on database interaction or use server-side scripting should be hosted on Amazon EC2. | Amazon EC2 |

**Amazon Glacier**

Usage patterns:

Organizations are using Amazon Glacier to support a number of use cases. These include archiving offsite enterprise information, media assets, research and scientific data, digital preservation and magnetic tape replacement.

Cost Model:

In normal use, Amazon Glacier has three pricing components: storage (per GB per month), data transfer out (per GB per month), and requests (per thousand UPLOAD and RETRIEVAL requests per month).

Note:

Amazon Glacier is designed with the expectation that retrievals are infrequent and unusual, and data will be stored for extended periods of time. You can retrieve up to 5 percent of your average monthly storage (prorated daily) for free each month. If you retrieve more than this amount of data in a month, you are charged an additional (per GB) retrieval fee.

Anti-patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | AWs Services |
| Rapidly changing data | Data that must be updated very frequently might be better served by a storage solution with lower read/write latencies, such as Amazon EBS, Amazon RDS, Amazon DynamoDB, or relational databases running on EC2. | Amazon EBS volumes  Amazon RDS  Amazon Dynamo DB  Amazon EC2 |
| Immediate Access | Data stored in Amazon Glacier is not available immediately. Retrieval jobs typically require 3–5 hours to complete, so if you need immediate access to your data, Amazon S3 is a better choice. | Amazon S3 |

**Amazon EBS**

Usage Patterns:

Amazon EBS is meant for data that changes relatively frequently and requires long-term persistence. Amazon EBS is well suited for use as the primary storage for a database or file system, or for any applications that require access to raw block-level storage.

Amazon EBS volume types:

* Amazon EBS Magnetic - well suited for infrequently accessed data
* Amazon EBS General Purpose (solid-state drive (SSD) backed) - well suited for boot volumes and small to medium databases.
* Amazon EBS Provisioned IOPS (SSD backed) - well suited for use with I/O-intensive workloads, relational databases, and NoSQL databases.

Cost model:

* Amazon EBS pricing has three components: provisioned storage, I/O requests, and snapshot storage.
* Amazon EBS Magnetic volumes are charged per GB-month of provisioned storage and per million I/O requests.
* Amazon EBS General Purpose (SSD) volumes are charged per GB-month of provisioned storage.
* Amazon EBS Provisioned IOPS (SSD) volumes are charged per GB-month of provisioned storage and per Provisioned IOPS–month.
* For all volume types, Amazon EBS snapshots are charged per GB-month of data stored. Amazon EBS snapshot copy is charged for the data transferred between regions and for the standard Amazon EBS snapshot charges in the destination region.
* It’s important to remember that for Amazon EBS volumes, you are charged for provisioned (allocated) storage, whether or not you actually use it. For Amazon EBS snapshots, you are charged only for storage actually used (consumed).

Note:

1. Amazon EBS snapshots are incremental and compressed, so the storage used in any snapshot is generally much less than the storage consumed for an Amazon EBS volume.
2. There is no charge for transferring information among the various AWS storage offerings (that is, an Amazon EC2 instance transferring information with Amazon EBS, Amazon S3, Amazon RDS, and so on) as long as the storage offerings are within the same AWS region.

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Services |
| Temporary storage | Consider using local instance store volumes for needs such as scratch disks, buffers, queues, and caches | Amazon Local  Instance Store |
| Highly durable storage | If you need very highly durable storage, use Amazon S3. Amazon S3 standard storage is designed for 99.999999999 percent annual durability per object. You can even decide to snapshot the EBS volumes. Such a snapshot then gets saved in Amazon S3, thus providing you the durability of Amazon S3. | Amazon S3 |
| Static data or web content | If your data doesn’t change that often, Amazon S3 might represent a more cost-effective and scalable solution for storing this fixed information. Also, web content served out of Amazon EBS requires a web server running on Amazon EC2; in contrast, you can deliver web content directly out of Amazon S3. | Amazon S3 |

**Amazon EC2 Instance Store**

Usage Patterns:

1. EC2 local instance store volumes are ideal for temporary storage of information that is continually changing, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.
2. Amazon EC2 instance storage is well-suited for this purpose. It consists of the virtual machine’s boot device (for instance store AMIs only), plus one or more additional volumes that are dedicated to the Amazon EC2 instance (for both Amazon EBS AMIs and instance store AMIs). This storage can only be used from a single Amazon EC2 instance during that instance's lifetime.
3. For high I/O and high storage, use Amazon EC2 instance storage targeted to these use cases. High I/O instances (the I2 family) provide instance store volumes backed by SSD and are ideally suited for many high-performance database workloads.
4. **Example** Applications include NoSQL databases like Cassandra and MongoDB, clustered databases, online transaction processing (OLTP) systems, and similar. High storage instances (the D2 family) support much higher storage density per Amazon EC2 instance, and are ideally suited for applications that benefit from high sequential I/O performance across very large datasets , applications include data warehouses, Hadoop/MapReduce storage nodes, parallel file systems, and similar.

Note:

* Some instance types, such as the micro instances (t1, t2) and the Compute Optimized C4 instances, use Amazon EBS storage only with no instance storage provided.
* Instances using Amazon EBS for the root device (in other words, that boot from Amazon EBS) don’t expose the instance store volumes by default. You can choose to expose the instance store volumes at instance launch time by specifying a block device mapping.
* Applications using instance storage for persistent data generally provide data durability through replication or by periodically copying data to durable storage.

Cost Model:

* The cost of an Amazon EC2 instance includes any local instance store volumes, if the instance type provides them. Although there is no additional charge for data storage on local instance store volumes,
* Data transferred to and from Amazon EC2 instance store volumes from other Availability Zones or outside of an Amazon EC2 region can incur data transfer charges, and additional charges apply for use of any persistent storage, such as Amazon S3, Amazon Glacier, Amazon EBS volumes, and Amazon EBS snapshots.

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Services |
| Persistent storage | If you need persistent virtual disk storage similar to a physical disk drive for files or other data that must persist longer than the lifetime of a single Amazon EC2 instance, Amazon EBS volumes or Amazon S3 are more appropriate | Amazon EC2  Amazon S3  Amazon EBS |
| Relational database storage | In most cases, relational databases require storage that persists beyond the lifetime of a single Amazon EC2 instance, making Amazon EBS volumes the natural choice. | Amazon EC2  Amazon EBS |
| Shared storage | Instance store volumes are dedicated to a single Amazon EC2 instance and cannot be shared with other systems or users. If you need storage that can be detached from one instance and attached to a different instance, or if you need the ability to share data easily, Amazon EFS, Amazon S3, or Amazon EBS volumes are the better choice | Amazon EFS  Amazon S3  Amazon EBS |
| Snapshots | If you need the convenience, long-term durability, availability, and the ability to share point-in-time disk snapshots, Amazon EBS volumes are a better choice. | Amazon EBS |

**AWS Storage Gateway**

Usage patterns:

Organizations are using AWS Storage Gateway to support a number of use cases includes corporate file sharing, enabling existing on-premises backup applications to store primary backups on Amazon S3, disaster recovery, and mirroring data to cloud-based compute resources and then later archiving it to Amazon Glacier.

Cost Model:

AWS Storage Gateway has the following pricing components: gateway usage (per gateway per month), snapshot storage usage (per GB per month), volume storage usage (per GB per month), virtual tape shelf storage (per GB per month), virtual tape library storage (per GB per month), retrieval from virtual tape shelf (per GB), and data transfer out (per GB per month).

**AWS Import/Export Snowball**

Usage Patterns:

1. In general, if loading your data over the Internet would take a week or more, you should consider using AWS Import/Export

Snowball.

1. AWS Import/Export Snowball is ideal for transferring anywhere from terabytes to many petabytes of data in and out of the AWS cloud securely, especially in cases where you don’t want to make expensive upgrades to your network infrastructure, frequently experience large backlogs of data, are in a physically isolated environment, or are in an area where high-speed Internet connections are not available or cost-prohibitive.
2. Common use cases include cloud migration, disaster recovery, data center decommission, and content distribution.
3. In decommission of a data center, many steps are involved to make sure valuable data is not lost, and Snowball can help ensure data is securely and cost-effectively transferred to AWS.
4. In a content distribution scenario, you might use Snowball appliances if you regularly receive or need to share large amounts of data with clients, customers, or business associates. Snowball appliances can be sent directly from AWS to client or customer locations.

Anti Patterns:

AWS Import/Export Snowball might not be the ideal solution if your data can be transferred over the Internet in less than one week.

Cost Model:

AWS Import/Export Snowball has three pricing components: service fee (per job), extra day charges as required (the first 10 days of onsite usage are free), and data transfer. For the destination storage, the standard Amazon S3 storage pricing applies.

**Amazon CloudFront**

Usage Patterns:

1. Amazon CloudFront is also commonly used to stream audio and video files to web browsers and mobile devices.
2. Amazon CloudFront is ideal for distribution of frequently accessed static content that benefits from edge delivery, such as popular website images, videos, media files or software downloads.
3. Amazon CloudFront can also be used to deliver dynamic web applications over HTTP.
4. If you need to remove an object from Amazon CloudFront edge-server caches before it expires, you can either invalidate the object or use object versioning to serve a different version of the object that has a different name.

Note:

It might be better to serve infrequently accessed data directly from the origin server, avoiding the additional cost of origin fetches for data that is not likely to be reused at the edge.

Cost Model:

* Amazon CloudFront has two pricing components: regional data transfer out (per GB) and requests (per 10,000).
* If you use an AWS service as the origin (for example, Amazon S3, Amazon EC2, Elastic Load Balancing, or others), data transferred from origin to edge locations (Amazon CloudFront “origin fetches”) will be free of charge.
* Data transfer out of Amazon CloudFront to your origin server will be billed at the “Regional Data Transfer Out of Origin” rates.
* CloudFront provides three different price classes according to where your content needs to be distributed. If you don’t need your content to be distributed globally, but only within certain locations such as the US and Europe, you can lower the prices you pay to deliver by choosing a price class that includes only these locations.
* CloudFront offers an optional reserved capacity plan that gives you the option to commit to a minimum monthly usage level for 12 months or longer and in turn receive a significant discount

**Amazon Simple Queue Service (SQS)**

Usage Patterns:

* Amazon SQS is ideally suited to any scenario where multiple application components must communicate and coordinate their work in a loosely coupled manner. This occurs particularly in producer-consumer scenarios where some components may work faster or slower than others, or where the number of interacting components changes with time or load.
* A classic use of Amazon SQS is to coordinate a multi-step processing pipeline, where each message is associated with a task that must be processed. Each task is described by an Amazon SQS message indicating the task to be done and a pointer to the task data in Amazon S3.
* Amazon SQS can serve as the “software glue” that enables components to communicate reliably without being tightly coupled or highly dependent upon synchronous operation, or on a fixed number of components.
* Use of the Amazon SQS queue enables the number of worker instances to scale up or down, and also enable the processing power of each single worker instance to scale up or down, to suit the total workload, without any application changes.

Cost Model:

Amazon SQS provides a free tier of service which provides 100,000 requests per month at no charge. Beyond the free tier, Amazon SQS pricing is based on number of requests (priced per 10,000 requests) and the amount of data transferred in and out (priced per GB per month).

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Services |
| Binary or large messages | Amazon SQS messages must be text, and can be a maximum of 64 KB in length. If the data you need to store in a queue exceeds this length, or is binary, it is best to use Amazon S3 or Amazon RDS to store the large or binary data, and store a pointer to the data in Amazon SQS | Amazon S3  Amazon RDS |
| Long-term storage | If message data must be stored for longer than 14 days, Amazon S3 or some other storage mechanism is more appropriate. | Amazon S3  Amazon Glacier |
| High-speed message queuing or very short tasks | If your application requires a very high-speed message send and receive response from a single producer or consumer, use of Amazon DynamoDB or a message-queuing system hosted on Amazon EC2 may be more appropriate | Amazon DynamoDB |

**Amazon Relational Database Service (RDS)**

Usage patterns:

* Amazon RDS is ideal for existing applications that rely on MySQL, Oracle, or SQL Server traditional relational database engines, Since Amazon RDS offers full compatibility and direct access to native database engines, most code, libraries, and tools designed for these databases should work unmodified with Amazon RDS
* Amazon RDS is also optimal for new applications with structured data that requires more sophisticated querying and joining capabilities than that provided by Amazon’s NoSQL database offering, Amazon DynamoDB

Cost Model:

Pricing for Amazon RDS is based on several factors: the DB instance hours (per hour), the amount of provisioned database storage (per GB-month and per million I/O requests), additional backup storage (per GB-month), and data transfer in / out (per GB per month).

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Services |
| Index and query-focused data | Many cloud-based solutions don’t require advanced features found in a relational database, such as joins and complex transactions. If your application is more oriented toward indexing and querying data, you may find Amazon DynamoDB to be more appropriate for your needs. | Amazon DynamoDB |
| Numerous BLOBs | While all of the database engines provided by Amazon RDS support binary large objects (BLOBs), if your application makes heavy use of them (audio files, videos, images, and so on), you may find Amazon S3 to be a better choice | Amazon S3 |
| Automated scalability | Amazon RDS provides pushbutton scaling. If you need fully automated scaling, Amazon DynamoDB may be a better choice | Amazon DynamoDB |
| Other database platforms | Amazon RDS provides a MySQL, Oracle, and SQL Server databases at this time. If you need another database platform (such as IBM DB2, Informix, PostgreSQL, or Sybase) you need to deploy a self-managed database on an Amazon EC2 instance by using a relational database AMI, or by installing database software on an Amazon EC2 instance. |  |
| Complete control | If your application requires complete, OS-level control of the database server, with full root or admin login privileges (for example, to install additional third-party software on the same server), a self managed database on Amazon EC2 may be a better match |  |

**Amazon DynamoDB**

Usage Patterns:

* Amazon DynamoDB is ideal for existing or new applications that need a flexible NoSQL database with low read and write latencies, and the ability to scale storage and throughput up or down as needed without code changes or downtime.
* Common use cases include: mobile apps, gaming, digital ad serving, live voting and audience interaction for live events, sensor networks, log ingestion, access control for web-based content, metadata storage for Amazon S3 objects, e- commerce shopping carts, and web session management. Many of these use cases require a highly available and scalable database because downtime or performance degradation has an immediate negative impact on an organization’s business.

Cost Model:

Amazon DynamoDB has three pricing components: provisioned throughput capacity (per hour), indexed data storage (per GB per month), data transfer in or out (per GB per month).

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Services |
| Prewritten application tied to a traditional relational database | If you are attempting to port an existing application to the AWS cloud, and need to continue using a relational database, you may elect to use either Amazon RDS (MySQL, Oracle, or SQL Server), or one of the many preconfigured Amazon EC2 database AMIs. You are also free to create your own Amazon EC2 instance, and install your own choice of database software | Amazon RDS |
| Joins and/or complex transactions | While many solutions are able to leverage Amazon DynamoDB to support their users, it’s possible that your application may require joins, complex transactions, and other relational infrastructure provided by traditional database platforms. If this is the case, you may want to explore Amazon RDS or Amazon EC2 with a self-managed database. | Amazon RDS |
| BLOB data | If you plan on storing large (greater than 64 KB) BLOB data, such as digital video, images, or music, you’ll want to consider Amazon S3. However, Amazon DynamoDB still has a role to play in this scenario, for keeping track of metadata (e.g., item name, size, date created, owner, location, and so on) about your binary objects. | Amazon S3 |
| Large data with low I/O rate | Amazon DynamoDB uses SSD drives and is optimized for workloads with a high I/O rate per GB stored. If you plan to store very large amounts of data that are infrequently accessed, other storage options, such as Amazon S3, may be a better choice | Amazon S3 |

**Amazon Elastic Cache**

Usage Patterns:

* ElastiCache improves application performance by storing critical pieces of data in memory for low-latency access. It is frequently used as a database front end in read-heavy applications, improving performance and reducing the load on the database by caching the results of I/O-intensive queries.
* It is also frequently used to manage web session data, to cache dynamically-generated web pages, and to cache results of computationally-intensive calculations, such as the output of recommendation engines
* For applications that need more complex data structures than strings, such as lists, sets, hashes, and sorted sets, the Redis engine is often used as an in-memory NoSQL database.

Cost Model:

ElastiCache has only a single pricing component: pricing is per cache node-hour consumed.

Anti-patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Service |
| Persistent data | If you need very fast access to data, but also need strong data durability (persistence), Amazon DynamoDB is probably a better choice. | Amazon DynamoDB |

**Amazon Redshift**

Usage Patterns:

Amazon Redshift is ideal for analyzing large datasets using your existing business intelligence tools.

Cost Model:

1. Amazon Redshift has three pricing components: data warehouse node hours, backup storage, and data transfer
2. Compute node hours are the total number of hours run across all compute nodes for the billing period.
3. Backup storage is the storage associated with automated and manual snapshots for an Amazon Redshift data warehouse cluster.
4. Increasing the backup retention period or taking additional snapshots increases the backup storage consumed by the Amazon Redshift data warehouse cluster.
5. There is no additional charge for backup storage up to 100% of your provisioned storage for an active data warehouse cluster. There is no data transfer charge for data transferred to or from Amazon Redshift outside of Amazon Virtual Private Cloud (Amazon VPC). Data transfer to or from Amazon Redshift in Amazon VPC accrues standard AWS data transfer charges

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS Services |
| OLTP workloads | Amazon Redshift is a column-oriented database suited to data warehouse and analytics, where queries are typically performed over very large datasets. If your application involves online transaction processing, a traditional row-based database system, such as Amazon RDS, is a better match. | Amazon RDS |
| BLOB data | If you plan on storing binary (e.g., video, pictures, or music), you’ll want to consider Amazon S3 | Amazon S3 |

**Databases on Amazon EC2**

Usage Patterns:

Running a relational database on Amazon EC2 and Amazon EBS is the ideal scenario for users whose application requires a specific traditional relational database not supported by Amazon RDS or for those users who require a maximum level of administrative control and configurability.

Cost Model:

1. The cost of running your own database on Amazon EC2 depends on the size and number of Amazon EC2 instances used to run your database, the size of the Amazon EBS volumes used for database storage, the amount of data transferred in and out of Amazon EC2, and in many cases, the license cost of the third-party database software.
2. Many open-source database packages use a no-cost license model; some commercial software vendors use the Amazon DevPay model; many others provide a bring-your-own-license model. Contact your database software vendor or Amazon Web Services to understand the license cost pricing model that applies.

Anti-Patterns:

|  |  |  |
| --- | --- | --- |
| Storage Need | Solution | Alternative AWS services |
| Index and query-focused data | Many cloud-based solutions don’t require advanced features found in a relational database, such as joins or complex transactions. If your application is more oriented toward indexing and querying data, you may find Amazon DynamoDB to be more appropriate for your needs, and significantly easier to manage. | Amazon DynamoDB |
| Numerous BLOBs | Many relational databases support BLOBs (audio files, videos, images, and so on). If your application makes heavy use of them, you may find Amazon S3 to be a better choice. You can use a database to manage the metadata. | Amazon S3 |
| Automatic scaling | Users of relational databases on AWS in many cases, leverage the scalability and elasticity of the underlying AWS platform, but this requires system administrators or DBAs to perform a manual or scripted task. If you need pushbutton scaling or fully-automated scaling, you may opt for another storage choice such as Amazon DynamoDB or Amazon RDS. | Amazon DynamoDB  Amazon RDS |
| MySQL, Oracle, SQL Server | If you are running a self-managed MySQL, Oracle, or SQL Server database on Amazon EC2, you should consider the automated backup, patching, Provisioned IOPS, replication, and pushbutton scaling features offered by a fully-managed Amazon RDS database. | Amazon RDS |