STUDY TIP: Focus on the following FAQs:

[Amazon Athena](https://aws.amazon.com/athena/faqs/) | [Amazon EMR](https://aws.amazon.com/emr/faqs/) | [Amazon Redshift](https://aws.amazon.com/redshift/faqs/) | [Amazon CloudSearch](https://aws.amazon.com/cloudsearch/faqs/) | [Amazon Kinesis Video Streams](https://aws.amazon.com/kinesis/video-streams/faqs/) | [Amazon Kinesis Data Streams](https://aws.amazon.com/kinesis/data-streams/faqs/) | [Amazon Kinesis Data Firehose](https://aws.amazon.com/kinesis/data-firehose/faqs/) | [Amazon Kinesis Data Analytics](https://aws.amazon.com/kinesis/data-analytics/faqs/)  | [Amazon ElasticSearch Service](https://aws.amazon.com/elasticsearch-service/faqs/) | [Amazon Managed Service for Kafka (MSK)](https://aws.amazon.com/msk/faqs/) | [Amazon Redshift](https://aws.amazon.com/redshift/faqs/) | [Amazon Quicksight](https://aws.amazon.com/quicksight/resources/faqs/) | [AWS Data Exchange](https://aws.amazon.com/data-exchange/faqs/) | [AWS Glue](https://aws.amazon.com/glue/faqs/) | [AWS Lake Formation](https://aws.amazon.com/lake-formation/faqs/) | [AWS Data Pipeline](https://aws.amazon.com/datapipeline/faqs/)

Collection

[Resilience in Amazon Kinesis Data Firehose](https://docs.aws.amazon.com/firehose/latest/dev/disaster-recovery-resiliency.html)

[Resilience in Amazon Kinesis Data Streams](https://docs.aws.amazon.com/streams/latest/dev/disaster-recovery-resiliency.html)

[KCL](https://docs.aws.amazon.com/streams/latest/dev/shared-fan-out-consumers.html)

[Kinesis Agent](https://docs.aws.amazon.com/streams/latest/dev/writing-with-agents.html)

[KPL](https://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-kpl.html)

[Kinesis Streams API](https://docs.aws.amazon.com/streams/latest/dev/developing-producers-with-sdk.html)

[Troubleshooting Amazon Kinesis Data Firehose](https://docs.aws.amazon.com/firehose/latest/dev/troubleshooting.html)

[Troubleshooting errors in AWS Glue](https://docs.aws.amazon.com/glue/latest/dg/glue-troubleshooting-errors.html)

[Troubleshooting migration tasks in DMS](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Troubleshooting.html)

[Troubleshooting Kinesis producers](https://docs.aws.amazon.com/streams/latest/dev/troubleshooting-producers.html)

[Troubleshooting Kinesis consumers](https://docs.aws.amazon.com/streams/latest/dev/troubleshooting-consumers.html)

 [Kinesis Data Streams](https://aws.amazon.com/kinesis/data-streams/faqs/)

* [Kinesis Data Streams pricing](https://aws.amazon.com/kinesis/data-streams/pricing/)

[Kinesis Data Firehose](https://aws.amazon.com/kinesis/data-firehose/faqs/)

* [Kinesis Data Firehose pricing](https://aws.amazon.com/kinesis/data-firehose/pricing/)

 [Kinesis Data Analytics](https://aws.amazon.com/kinesis/data-analytics/faqs/)

 [Kinesis Video Streams](https://aws.amazon.com/kinesis/video-streams/faqs/)

[AWS Glue](https://aws.amazon.com/glue/faqs/)

* [AWS Glue pricing](https://aws.amazon.com/glue/pricing/)

[AWS DMS](https://aws.amazon.com/dms/faqs/)

* [CDC](https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Task.CDC.html)
* [AWS DMS pricing](https://aws.amazon.com/dms/pricing/)
* [AWS Schema Conversion Tool](https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_Welcome.html)

Ingestion system bottlenecks

* [Kinesis Data Streams resharding, scaling, and parallel processing](https://docs.aws.amazon.com/streams/latest/dev/kinesis-record-processor-scaling.html)
* [Troubleshooting high target latency on AWS DMS task](https://aws.amazon.com/premiumsupport/knowledge-center/dms-high-target-latency/)
* [KPL retries and rate limiting](https://docs.aws.amazon.com/streams/latest/dev/kinesis-producer-adv-retries-rate-limiting.html)
*  [Change data capture (CDC)](https://aws.amazon.com/blogs/database/aws-dms-now-supports-native-cdc-support/)
* [Amazon SQS](https://aws.amazon.com/sqs/faqs/)
* [Apache Kafka](https://aws.amazon.com/msk/what-is-kafka/)
* [Amazon MSK](https://aws.amazon.com/msk/faqs/)
* [Amazon SNS](https://aws.amazon.com/sns/faqs/)
* [Amazon Kinesis](https://aws.amazon.com/kinesis/)
* [Amazon DynamoDB Streams](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html)

 [DynamoDB](https://aws.amazon.com/dynamodb/faqs/)

* [Partition key and sort key](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html)
* [Consistency models](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadConsistency.html)
* [WCUs and RCUs](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.ReadWriteCapacityMode.html)
* [DynamoDB global tables](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GlobalTables.html)
* [DynamoDB Streams](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html)



[Amazon RDS](https://aws.amazon.com/rds/faqs/)



[Amazon S3](https://aws.amazon.com/s3/faqs/)



[Amazon Redshift](https://aws.amazon.com/redshift/faqs/)



[Amazon ElastiCache](https://aws.amazon.com/elasticache/faqs/)

 [Amazon ElastiCache](https://aws.amazon.com/elasticache/faqs/)

 [Amazon DAX](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX.html)

[Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html)

* [Amazon S3 storage classes](https://aws.amazon.com/s3/storage-classes/)

 [Amazon S3 Glacier](https://aws.amazon.com/glacier/faqs/)

 [Data lakes vs. data warehouses](https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/)

 [Amazon Redshift compute nodes](https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html)

[**Best Practices for Designing and Architecting with DynamoDB**](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/best-practices.html)**.**

[**Best Practices for Designing and Architecting with DynamoDB**](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/best-practices.html)**.**

For sample code that demonstrates how to compress such messages in DynamoDB, see:

* Example: [Handling Binary Type Attributes Using the AWS SDK for Java Document API](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/JavaDocumentAPIBinaryTypeExample.html)
* Example: [Handling Binary Type Attributes Using the AWS SDK for .NET Low-Level API](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/LowLevelDotNetBinaryTypeExample.html)

Compression

* [Raw Encoding](https://docs.aws.amazon.com/redshift/latest/dg/c_Raw_encoding.html)
* [AZ64 Encoding](https://docs.aws.amazon.com/redshift/latest/dg/az64-encoding.html)
* [Byte-Dictionary Encoding](https://docs.aws.amazon.com/redshift/latest/dg/c_Byte_dictionary_encoding.html)
* [Delta Encoding](https://docs.aws.amazon.com/redshift/latest/dg/c_Delta_encoding.html)
* [LZO Encoding](https://docs.aws.amazon.com/redshift/latest/dg/lzo-encoding.html)
* [Mostly Encoding](https://docs.aws.amazon.com/redshift/latest/dg/c_MostlyN_encoding.html)
* [Runlength Encoding](https://docs.aws.amazon.com/redshift/latest/dg/c_Runlength_encoding.html)
* [Text255 and Text32k Encodings](https://docs.aws.amazon.com/redshift/latest/dg/c_Text255_encoding.html)
* [Zstandard Encoding](https://docs.aws.amazon.com/redshift/latest/dg/zstd-encoding.html)

 [Amazon Redshift](https://docs.aws.amazon.com/redshift/latest/mgmt/overview.html)

* [Architecture](https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html)
* [Sort keys](https://docs.aws.amazon.com/redshift/latest/dg/t_Sorting_data.html)
* [Distribution styles](https://docs.aws.amazon.com/redshift/latest/dg/t_Distributing_data.html)
* [Loading data best practices](https://docs.aws.amazon.com/redshift/latest/dg/t_Loading_data.html)
* [Compression encodings](https://docs.aws.amazon.com/redshift/latest/dg/c_Compression_encodings.html)

[Amazon DynamoDB](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html)

* [GSIs](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html)
* [LSIs](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/LSI.html)
* [Partitions](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.Partitions.html)
* [Throttling](https://aws.amazon.com/premiumsupport/knowledge-center/throttled-ddb/)

[Amazon S3](https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html)

* [Compression techniques](https://docs.aws.amazon.com/AmazonS3/latest/dev/optimizing-performance.html)
* [Optimal file size](https://docs.aws.amazon.com/AmazonS3/latest/dev/optimizing-performance.html)

 See more Amazon Redshift loading best practices [here](https://docs.aws.amazon.com/redshift/latest/dg/c_best-practices-compress-data-files.html).

For more information on additional considerations for transitioning objects to another storage class, see [Transitioning Objects Using Amazon S3 Lifecycle](https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-transition-general-considerations.html).

 [Amazon S3](https://aws.amazon.com/s3/faqs/)

* [Lifecycle policies](https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html)
* [Cross-region replication rules](https://docs.aws.amazon.com/AmazonS3/latest/dev/replication.html)

[Amazon Redshift](https://aws.amazon.com/redshift/faqs/)

* [Backup and restore](https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-snapshots.html)

[Amazon DynamoDB](https://aws.amazon.com/dynamodb/faqs/)

* [Backup and restore](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html)

[Amazon RDS](https://aws.amazon.com/rds/faqs/)

* [Backup and restore](https://aws.amazon.com/rds/details/backup/)

For more information on Amazon Redshift's backup and restore methods, see [Working with Snapshots](https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-snapshots.html).

For more information, see [DynamoDB backup and restore](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html).

For more information about creating a DB snapshot, see [Creating a Database Snapshot](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_CreateSnapshot.html).

For more information on backing up and restoring with Amazon RDS, see [RDS backup and restore](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_CommonTasks.BackupRestore.html).

**Alternatively to AWS Glue, you can also host a Hive metastore on Amazon RDS.**

If you need an alternative to AWS Glue, due to the [limitations listed in the documentation](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive-metastore-glue.html#emr-hive-glue-considerations-hive) or if you want to integrate with other open-source applications such as Apache Ranger or Apache Atlas, then consider hosting your Hive metastore on Amazon RDS.

Keep in mind that your Hive metastore is a single point of failure. Amazon RDS doesn’t automatically replicate databases, so you should enable replication when using Amazon RDS to avoid any data loss in the event of failure.

For more information on how to set this up, see [Migrate and deploy your Apache Hive metastore on Amazon EMR](https://aws.amazon.com/blogs/big-data/migrate-and-deploy-your-apache-hive-metastore-on-amazon-emr/).

 [AWS Glue Data Catalog](https://docs.aws.amazon.com/glue/latest/dg/populate-data-catalog.html)

 [Hive metastore](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-metastore-external-hive.html)

 [Amazon Athena](https://docs.aws.amazon.com/athena/latest/ug/glue-athena.html)

 [Amazon EMR](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark-glue.html)

 [Amazon Redshift spectrum](https://docs.aws.amazon.com/redshift/latest/dg/c-spectrum-external-schemas.html)

See the full process of how to load data from Amazon EMR into Amazon Redshift [here](https://docs.aws.amazon.com/redshift/latest/dg/loading-data-from-emr.html).

 [Amazon EMR](https://aws.amazon.com/emr/faqs/)

* [Working with file systems in Amazon EMR](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-file-systems.html)
* [Amazon EMR pricing](https://aws.amazon.com/emr/pricing/?nc=sn&loc=4)
* [Amazon EMR integrations](https://aws.amazon.com/emr/features/)
* [Amazon EMR applications](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-ha-applications.html)

 [AWS Glue](https://aws.amazon.com/glue/faqs/)

* [AWS Glue pricing](https://aws.amazon.com/glue/pricing/)
* [AWS Glue ETL](https://docs.aws.amazon.com/glue/latest/dg/author-job.html)

 [Apache Spark](https://aws.amazon.com/big-data/what-is-spark/)

[Amazon Kinesis Data Analytics](https://aws.amazon.com/kinesis/data-analytics/faqs/)

* [Kinesis Data Analytics pricing](https://aws.amazon.com/kinesis/data-analytics/pricing/?nc=sn&loc=3)
* [Kinesis Data Analytics ETL](https://aws.amazon.com/blogs/big-data/preprocessing-data-in-amazon-kinesis-analytics-with-aws-lambda/)
* [Using AWS Lambda with Amazon Kinesis](https://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html).

[Amazon EMR](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-overview.html)

* [EMR Cluster configuration guidelines and best practices](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-instances-guidelines.html)
* [Amazon EMR scalability](https://aws.amazon.com/blogs/big-data/best-practices-for-resizing-and-automatic-scaling-in-amazon-emr/)
* [Amazon EMR concurrency](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-concurrent-steps.html)
* [Amazon EMR troubleshooting a failed cluster](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-troubleshoot-failed.html)
* [Creating an EMR cluster with instance fleets or uniform instance groups](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-instance-group-configuration.html)
* [Best practices for running Apache Spark applications using EC2 Spot Instances on Amazon EMR](https://aws.amazon.com/blogs/big-data/best-practices-for-running-apache-spark-applications-using-amazon-ec2-spot-instances-with-amazon-emr/)

see [Performing Complex ETL Activities Using Workflows in AWS Glue](https://docs.aws.amazon.com/glue/latest/dg/orchestrate-using-workflows.html).

[Amazon EMR](https://aws.amazon.com/emr/faqs/)

* [Respond to State Changes on Amazon EMR Clusters with Amazon CloudWatch Events](https://aws.amazon.com/blogs/big-data/respond-to-state-changes-on-amazon-emr-clusters-with-amazon-cloudwatch-events/)
* [Monitor metrics with Amazon CloudWatch](https://docs.aws.amazon.com/emr/latest/ManagementGuide/UsingEMR_ViewingMetrics.html)
* [View and monitor an EMR cluster](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-manage-view.html)
* [Viewing log files in Amazon EMR](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-manage-view-web-log-files.html)
* [Logging Amazon EMR API Calls in AWS CloudTrail](https://docs.aws.amazon.com/emr/latest/ManagementGuide/logging_emr_api_calls.html)
* [Viewing log files in Amazon EMR](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-manage-view-web-log-files.html)

[AWS Step Functions](https://aws.amazon.com/step-functions/)

* [Orchestrate Apache Spark applications using AWS Step Functions and Apache Livy](https://aws.amazon.com/blogs/big-data/orchestrate-apache-spark-applications-using-aws-step-functions-and-apache-livy/)
* [Using Step Functions to Orchestrate Amazon EMR Workloads](https://aws.amazon.com/blogs/aws/new-using-step-functions-to-orchestrate-amazon-emr-workloads/)
* [Orchestrate Amazon Redshift-Based ETL workflows with AWS Step Functions and AWS Glue](https://aws.amazon.com/blogs/big-data/orchestrate-amazon-redshift-based-etl-workflows-with-aws-step-functions-and-aws-glue/)

[AWS Glue workflows](https://docs.aws.amazon.com/glue/latest/dg/workflows_overview.html)

You should have familiarity with these [AWS purpose-built analytics services](https://aws.amazon.com/big-data/datalakes-and-analytics/).

**Athena**

<https://aws.amazon.com/athena/resources/>

<https://docs.aws.amazon.com/athena/latest/ug/what-is.html>

<https://aws.amazon.com/athena/faqs/>

**Elasticsearch**

<https://aws.amazon.com/elasticsearch-service/resources/?whats-new-ess.sort-by=item.additionalFields.postDateTime&whats-new-ess.sort-order=desc>

<https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/what-is-amazon-elasticsearch-service.html>

<https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/aes-bp.html>

<https://aws.amazon.com/elasticsearch-service/faqs/>

**EMR**

<https://aws.amazon.com/emr/resources/>

<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html>

<https://aws.amazon.com/emr/faqs/>

**Kinesis Analytics**

<https://aws.amazon.com/kinesis/data-analytics/resources/>

<https://docs.aws.amazon.com/kinesisanalytics/latest/dev/what-is.html>

<https://aws.amazon.com/kinesis/data-analytics/faqs/>

**Kinesis Firehose**

<https://aws.amazon.com/kinesis/data-firehose/resources/>

<https://docs.aws.amazon.com/firehose/latest/dev/what-is-this-service.html>

<https://aws.amazon.com/kinesis/data-firehose/faqs/>

**Kinesis Data Streams**

<https://aws.amazon.com/kinesis/data-streams/resources/>

<https://docs.aws.amazon.com/streams/latest/dev/introduction.html>

<https://aws.amazon.com/kinesis/data-streams/faqs/>

**Kinesis Video Streams**

<https://aws.amazon.com/kinesis/video-streams/resources/>

<https://docs.aws.amazon.com/kinesisvideostreams/latest/dg/what-is-kinesis-video.html>

<https://aws.amazon.com/kinesis/video-streams/faqs/>

<https://aws.amazon.com/kinesis/video-streams/resources/>

**Redshift**

<https://docs.aws.amazon.com/redshift/latest/gsg/getting-started.html>

<https://docs.aws.amazon.com/redshift/latest/mgmt/welcome.html>

<https://docs.aws.amazon.com/redshift/latest/dg/welcome.html>

<https://aws.amazon.com/redshift/faqs/>

**Sagemaker**

<https://aws.amazon.com/sagemaker/developer-resources/>

<https://docs.aws.amazon.com/sagemaker/latest/dg/gs.html>

<https://aws.amazon.com/sagemaker/faqs/>

In addition to understanding how each of these services may be used in an analytics solution, you should know where and why [**Amazon S3**](https://aws.amazon.com/s3/developer-resources/), [**Amazon EC2**](https://aws.amazon.com/ec2/instance-types/)**,**[**AWS Glue**](https://aws.amazon.com/glue/resources/)**,**and [**AWS Lambda**](https://docs.aws.amazon.com/lambda/latest/dg/lambda-services.html) may be used as part of your analysis solution, in conjunction with their uses in data collection, storage, and processing phases.

The whitepaper [Big Data Analytics Options on AWS](https://d1.awsstatic.com/whitepapers/Big_Data_Analytics_Options_on_AWS.pdf) is a good reference for starting to understand the operational characteristics of these services including.

Pricing details at <https://aws.amazon.com/athena/pricing/>

Pricing details at <https://aws.amazon.com/elasticsearch-service/pricing/>

Pricing details at [Amazon EMR Pricing](https://aws.amazon.com/emr/pricing/).

Pricing details at [Amazon SageMaker pricing](https://aws.amazon.com/sagemaker/pricing/)

**Amazon Elasticsearch Service (Amazon ES)**

Performance depends on multiple factors including**:**

1. [**instance type**](https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/aes-supported-instance-types.html)
2. **workload**
3. **index**
4. [**number of shards** used](https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/sizing-domains.html#aes-bp-sharding)
5. **read replica** configuration
6. [**storage**configuration (instance or EBS)](https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-createupdatedomains.html#es-createdomain-configure-ebs)

The [Sizing Amazon ES Domains](https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/sizing-domains.html)page in the developer guide discusses factors for deciding how to size your Amazon ES domain.

**EMR**

For more information about Amazon EC2 instance specifications, see [Amazon EC2 Instance Types](http://aws.amazon.com/ec2/instance-types/).

Use the guidance in the [Cluster Configuration Guidelines and Best Practices](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-instances-guidelines.html) section of the management guide to become familiar with criteria for choosing **instance types**, **purchasing options**, and **storage** to provision for each node type in an Amazon EMR cluster.

The [Plan and Configure Clusters](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan.html) page within the Amazon EMR Management Guide highlights the choices that you need to make based on the data that you're processing and requirements for cost, speed, capacity, availability, security, and manageability.

Tutorial: [Optimizing Amazon EMR clusters for cost and scale](https://aws.amazon.com/getting-started/tutorials/optimize-amazon-emr-clusters-with-ec2-spot/)

**Amazon Kinesis**

Kinesis data streams performance

Choose throughput capacity in terms of shards. The [enhanced fan-out option](https://docs.aws.amazon.com/streams/latest/dev/introduction-to-enhanced-consumers.html) can improve performance by increasing the throughput available to each individual consumer.

Kinesis data firehose performance: [batch size or batch interval](https://docs.aws.amazon.com/firehose/latest/APIReference/API_PutRecordBatch.html)

Kinesis data analytics performance:  [use the InputParallelism parameter](https://docs.aws.amazon.com/kinesisanalytics/latest/dev/input-parallelism.html)

**Amazon Redshift**

Amazon Redshift uses a [variety of innovations](https://docs.aws.amazon.com/redshift/latest/dg/c_challenges_achieving_high_performance_queries.html) to obtain **very high performance**

Apply the tips outlined in [**Amazon Redshift Best Practices for Designing Tables**](https://docs.aws.amazon.com/redshift/latest/dg/c_designing-tables-best-practices.html)to maximize query efficiency.

Loading very large datasets can consume a lot of computing resources, and **how your data is loaded can also affect query performance**. Follow [best practices for loading data](https://docs.aws.amazon.com/redshift/latest/dg/c_loading-data-best-practices.html) efficiently.

You can [view performance data in the Amazon Redshift console](https://docs.aws.amazon.com/redshift/latest/mgmt/performance-metrics-console.html), which includes information about cluster and query performance. Additionally, you can create alarms on cluster metrics directly from the Amazon Redshift console.

The [Sizing Amazon ES Domains](https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/sizing-domains.html)page in the developer guide discusses factors for deciding how to size your Amazon ES domain.

Use the guidance in the [Cluster Configuration Guidelines and Best Practices](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-instances-guidelines.html) section of the management guide to become familiar with criteria for choosing **instance types**, **purchasing options**, and **storage** to provision for each node type in an Amazon EMR cluster.

The [Plan and Configure Clusters](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan.html) section within the Amazon EMR management guide highlights other choices that you make based on the data that you're processing and requirements for cost, speed, capacity, availability, security, and manageability.

Kinesis Data Streams performance

* Choose throughput capacity in terms of shards. The [enhanced fan-out option](https://docs.aws.amazon.com/streams/latest/dev/introduction-to-enhanced-consumers.html) can improve performance by increasing the throughput available to each individual consumer.
* **Amazon Redshift**
* Amazon Redshift uses a [variety of innovations](https://docs.aws.amazon.com/redshift/latest/dg/c_challenges_achieving_high_performance_queries.html) to obtain **very high performance** on data sets ranging in size from **hundreds of gigabytes to a petabyte or more**.

It uses [**columnar storage**](https://docs.aws.amazon.com/redshift/latest/dg/c_columnar_storage_disk_mem_mgmnt.html), **data compressio**n, and **zone maps** to reduce the amount of I/O needed to perform queries.

Apply the tips outlined in [**Amazon Redshift Best Practices for Designing Tables**](https://docs.aws.amazon.com/redshift/latest/dg/c_designing-tables-best-practices.html)to maximize query efficiency.

[Factors that can affect query performance](https://docs.aws.amazon.com/redshift/latest/dg/c-query-performance.html) include:

* **data**characteristics
* **cluster** configuration
* **database**operations

Loading very large datasets can consume a lot of computing resources, and **how your data is loaded can also affect query performance**. Follow [best practices for loading data](https://docs.aws.amazon.com/redshift/latest/dg/c_loading-data-best-practices.html) efficiently.

You can [view performance data in the Amazon Redshift console](https://docs.aws.amazon.com/redshift/latest/mgmt/performance-metrics-console.html), which includes information about cluster and query performance. Additionally, you can create alarms on cluster metrics directly from the Amazon Redshift console.

**Amazon sagemaker**

Amazon SageMaker hosting automatically scales to the performance needed for your application using [Application Auto Scaling](https://docs.aws.amazon.com/sagemaker/latest/dg/endpoint-auto-scaling.html). Find [best practices for configuring automatic scaling here](https://docs.aws.amazon.com/sagemaker/latest/dg/endpoint-auto-scaling-considerations.html).

By using[Amazon SageMaker Elastic Inference (EI)](https://docs.aws.amazon.com/sagemaker/latest/dg/ei.html), you can speed up the throughput and decrease the latency of getting real-time inferences from your deep learning models that are deployed as Amazon SageMaker hosted models.

The [Sizing Amazon ES Domains](https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/sizing-domains.html#aes-bp-instances) page of the developer guide provides general recommendations for calculating what you need.

Best practices for resizing a cluster are highlighted in [this blog post](https://aws.amazon.com/blogs/big-data/best-practices-for-resizing-and-automatic-scaling-in-amazon-emr/) within the Big Data blog.

Tutorial: [Optimizing Amazon EMR clusters for cost and scale](https://aws.amazon.com/getting-started/tutorials/optimize-amazon-emr-clusters-with-ec2-spot/)

**Amazon redshift**

You can easily [change the number or type of nodes](https://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html#rs-about-clusters-and-nodes) in your data warehouse as your performance or capacity needs change.

You can [use elastic resize to scale your cluster](https://docs.aws.amazon.com/redshift/latest/mgmt/rs-resize-tutorial.html) by changing the number of nodes. Or, you can use [classic resize](https://docs.aws.amazon.com/redshift/latest/mgmt/rs-resize-tutorial.html#classic-resize) to scale the cluster by specifying a different node type.

With the [Concurrency Scalin](https://docs.aws.amazon.com/redshift/latest/dg/concurrency-scaling.html)g feature, you can support virtually unlimited concurrent users and concurrent queries, with consistently fast query performance.

Amazon SageMaker hosting automatically scales to the performance needed for your application using [Application Auto Scalin](https://docs.aws.amazon.com/sagemaker/latest/dg/endpoint-auto-scaling.html)g.

When configuring automatic scaling, consider the following[general guidelines](https://docs.aws.amazon.com/sagemaker/latest/dg/endpoint-auto-scaling-considerations.html).

**Interfaces**

Amazon ES supports many of the commonly used open-source **Elasticsearch APIs**. The developer guide includes a list of [supported Elasticsearch operations](https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/aes-supported-es-operations.html).

[**Apache Hive**](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive.html)

[**Apache Pig**](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-pig.html)

[**Apache Spark**](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-spark.html)

[Apache HBase](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hbase.html)

[Presto](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-presto.html)

[Kinesis Connector](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-kinesis.html)

AWS provides the ability to quickly **move large amounts of data from Amazon S3 to HDFS**, **from HDFS to Amazon S3**, and **between Amazon S3 buckets** using Amazon EMR’s [S3DistCp](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/UsingEMR_s3distcp.html).

You can use the [EMR File System (EMRFS)](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-fs.html) and [enable Amazon S3 server-side and client-side encryption](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-emrfs-encryption.html). When you use EMRFS, a metadata store is transparently built-in DynamoDB to help manage the interactions with Amazon S3 and allows you to have multiple EMR clusters easily use the same EMRFS metadata and storage on Amazon S3.