Amazon EMR FAQs

## 

## **General**

Q: What is Amazon EMR?

Amazon EMR is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data. It utilizes a hosted Hadoop framework running on the web-scale infrastructure of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3).

Q: What can I do with Amazon EMR?

Using Amazon EMR, you can instantly provision as much or as little capacity as you like to perform data-intensive tasks for applications such as web indexing, data mining, log file analysis, machine learning, financial analysis, scientific simulation, and bioinformatics research. Amazon EMR lets you focus on crunching or analyzing your data without having to worry about time-consuming set-up, management or tuning of Hadoop clusters or the compute capacity upon which they sit.

Amazon EMR is ideal for problems that necessitate the fast and efficient processing of large amounts of data. The web service interfaces allow you to build processing workflows, and programmatically monitor progress of running clusters. In addition, you can use the simple web interface of the AWS Management Console to launch your clusters and monitor processing-intensive computation on clusters of Amazon EC2 instances.

Q: Who can use Amazon EMR?

Anyone who requires simple access to powerful data analysis can use Amazon EMR. You don’t need any software development experience to experiment with several sample applications available in the [Developer Guide](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/) and on [the AWS Big Data Blog](https://aws.amazon.com/blogs/big-data/).

Q: What can I do with Amazon EMR that I could not do before?

Amazon EMR significantly reduces the complexity of the time-consuming set-up, management. and tuning of Hadoop clusters or the compute capacity upon which they sit. You can instantly spin up large Hadoop clusters which will start processing within minutes, not hours or days. When your cluster finishes its processing, unless you specify otherwise, it will be automatically terminated so you are not paying for resources you no longer need.

Using this service you can quickly perform data-intensive tasks for applications such as web indexing, data mining, log file analysis, machine learning, financial analysis, scientific simulation, and bioinformatics research.

As a software developer, you can also develop and run your own more sophisticated applications, allowing you to add functionality such as scheduling, workflows, monitoring, or other features.

Q: What is the data processing engine behind Amazon EMR?

Amazon EMR uses [Apache Hadoop](http://hadoop.apache.org/) as its distributed data processing engine. Hadoop is an open source, Java software framework that supports data-intensive distributed applications running on large clusters of commodity hardware. Hadoop implements a programming model named “MapReduce,” where the data is divided into many small fragments of work, each of which may be executed on any node in the cluster. This framework has been widely used by developers, enterprises and startups and has proven to be a reliable software platform for processing up to petabytes of data on clusters of thousands of commodity machines.

Q: What is Apache Spark?

[Apache Spark](https://aws.amazon.com/big-data/what-is-spark/)TM is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size. Amazon EMR is the best place to deploy Apache Spark in the cloud, because it combines the integration and testing rigor of commercial Hadoop & Spark distributions with the scale, simplicity, and cost effectiveness of the cloud. It allows you to launch Spark clusters in minutes without needing to do node provisioning, cluster setup, Spark configuration, or cluster tuning. Learn [more about Spark](https://aws.amazon.com/big-data/what-is-spark/) and [Spark on Amazon EMR](https://aws.amazon.com/emr/details/spark.html/?cq_ck=1523458964032).

Q: What is Presto?

[Presto](https://aws.amazon.com/big-data/what-is-presto/) (or PrestoDB) is an open source, distributed SQL query engine, designed from the ground up for fast analytic queries against data of any size. With Amazon EMR, you can launch Presto clusters in minutes without needing to do node provisioning, cluster setup, Presto configuration, or cluster tuning. EMR enables you to provision one, hundreds, or thousands of compute instances in minutes. Learn [more about Presto](https://aws.amazon.com/big-data/what-is-presto/) and [Presto on Amazon EMR](https://aws.amazon.com/emr/details/presto.html/?cq_ck=1523460575370).

Q: What is an Amazon EMR cluster?

Amazon EMR historically referred to an Amazon EMR cluster (and all processing steps assigned to it) as a "cluster". Every cluster has a unique identifier that starts with "j-".

Q: What is a cluster step?

A cluster step is a user-defined unit of processing, mapping roughly to one algorithm that manipulates the data. A step is a Hadoop MapReduce application implemented as a Java jar or a streaming program written in Java, Ruby, Perl, Python, PHP, R, or C++. For example, to count the frequency with which words appear in a document, and output them sorted by the count, the first step would be a MapReduce application which counts the occurrences of each word, and the second step would be a MapReduce application which sorts the output from the first step based on the counts.

Q: What are different cluster states?

STARTING – The cluster provisions, starts, and configures EC2 instances.  
BOOTSTRAPPING – Bootstrap actions are being executed on the cluster.  
RUNNING – A step for the cluster is currently being run.  
WAITING – The cluster is currently active, but has no steps to run.  
TERMINATING - The cluster is in the process of shutting down.  
TERMINATED - The cluster was shut down without error.  
TERMINATED\_WITH\_ERRORS - The cluster was shut down with errors.

Q: What are different step states?

PENDING – The step is waiting to be run.  
RUNNING – The step is currently running.  
COMPLETED – The step completed successfully.  
CANCELLED – The step was cancelled before running because an earlier step failed or cluster was terminated before it could run.  
FAILED – The step failed while running.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Launching a Cluster**

Q: How can I access Amazon EMR?

You can access Amazon EMR by using the AWS Management Console, Command Line Tools, SDKS, or the EMR API.

Q: How can I launch a cluster?

You can launch a cluster through the AWS Management Console by filling out a simple cluster request form. In the request form, you specify the name of your cluster, the location in Amazon S3 of your input data, your processing application, your desired data output location, and the number and type of Amazon EC2 instances you’d like to use. Optionally, you can specify a location to store your cluster log files and SSH Key to login to your cluster while it is running. Alternatively, you can launch a cluster using the RunJobFlow API or using the ‘create’ command in the Command Line Tools.

Q: How can I get started with Amazon EMR?

To sign up for Amazon EMR, click the “Sign Up Now” button on the Amazon EMR detail page [http://aws.amazon.com/elasticmapreduce](https://aws.amazon.com/emr/). You must be signed up for Amazon EC2 and Amazon S3 to access Amazon EMR; if you are not already signed up for these services, you will be prompted to do so during the Amazon EMR sign-up process. After signing up, please refer to the [Amazon EMR documentation](https://aws.amazon.com/documentation/emr/), which includes our Getting Started Guide – the best place to get going with the service.

Q: How can I terminate a cluster?

At any time, you can terminate a cluster via the AWS Management Console by selecting a cluster and clicking the “Terminate” button. Alternatively, you can use the TerminateJobFlows API. If you terminate a running cluster, any results that have not been persisted to Amazon S3 will be lost and all Amazon EC2 instances will be shut down.

Q: Does Amazon EMR support multiple simultaneous cluster?

Yes. At any time, you can create a new cluster, even if you’re already running one or more clusters.

Q: How many clusters can I run simultaneously?

You can start as many clusters as you like. You are limited to 20 instances across all your clusters. If you need more instances, complete the [Amazon EC2 instance request form](http://aws.amazon.com/contact-us/ec2-request) and your use case and instance increase will be considered. If your Amazon EC2 limit has been already raised, the new limit will be applied to your Amazon EMR clusters.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Developing**

Q: Where can I find code samples?

Check out the sample code in these [Articles and Tutorials](http://aws.amazon.com/articles/Elastic-MapReduce).

Q: How do I develop a data processing application?

You can develop a data processing job on your desktop, for example, using Eclipse or NetBeans plug-ins such as IBM MapReduce Tools for Eclipse (<http://www.alphaworks.ibm.com/tech/mapreducetools>). These tools make it easy to develop and debug MapReduce jobs and test them locally on your machine. Additionally, you can develop your cluster directly on Amazon EMR using one or more instances.

Q: What is the benefit of using the Command Line Tools or APIs vs. AWS Management Console?

The Command Line Tools or APIs provide the ability to programmatically launch and monitor progress of running clusters, to create additional custom functionality around clusters (such as sequences with multiple processing steps, scheduling, workflow, or monitoring), or to build value-added tools or applications for other Amazon EMR customers. In contrast, the AWS Management Console provides an easy-to-use graphical interface for launching and monitoring your clusters directly from a web browser.

Q: Can I add steps to a cluster that is already running?

Yes. Once the job is running, you can optionally add more steps to it via the AddJobFlowSteps API. The AddJobFlowSteps API will add new steps to the end of the current step sequence. You may want to use this API to implement conditional logic in your cluster or for debugging.

Q: Can I run a persistent cluster?

Yes. Amazon EMR clusters that are started with the –alive flag will continue until explicitly terminated. This allows customers to add steps to a cluster on demand. You may want to use this to debug your application without having to repeatedly wait for cluster startup. You may also use a persistent cluster to run a long-running data warehouse cluster. This can be combined with data warehouse and analytics packages that runs on top of Hadoop such as Hive and Pig.

Q: Can I be notified when my cluster is finished?

You can sign up for up Amazon SNS and have the cluster post to your SNS topic when it is finished. You can also view your cluster progress on the AWS Management Console or you can use the Command Line, SDK, or APIs get a status on the cluster.

Q: What programming languages does Amazon EMR support?

You can use Java to implement Hadoop custom jars. Alternatively, you may use other languages including Perl, Python, Ruby, C++, PHP, and R via Hadoop Streaming. Please refer to the [Developer’s Guide](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/) for instructions on using Hadoop Streaming.

Q: What OS versions are supported with Amazon EMR?

Amazon EMR creates cluster instances using an [Amazon Linux Amazon Machine Image (AMI)](https://aws.amazon.com/amazon-linux-ami/) that is optimized for Amazon EMR. The AMI contains the Amazon Linux operating system, other software, and configurations required for each instance to host your cluster applications. As an alternative, you can specify a custom AMI that you create based on the Amazon Linux AMI. This allows you to perform sophisticated pre-configuration for virtually any application. For more information, see [Using a Custom AMI](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-custom-ami.html).

Q: Can I view the Hadoop UI while my cluster is running?

Yes. Please refer to the [Hadoop UI](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/UsingtheHadoopUserInterface.html) section in the Developer’s Guide for instructions on how to access the Hadoop UI.

Q: Does Amazon EMR support third-party software packages?

Yes. The recommended way to install third-party software packages on your cluster is to use Bootstrap Actions. Alternatively you can package any third party libraries directly into your Mapper or Reducer executable. You can also upload statically compiled executables using the Hadoop distributed cache mechanism.

Q: Which Hadoop versions does Amazon EMR support?

For the latest versions supported by Amazon EMR, please [reference the documentation](http://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/emr-release-components.html#d0e506).

Q: Does Amazon contribute Hadoop improvements to the open source community?

Yes. Amazon EMR is active with the open source community and contributes many fixes back to the Hadoop source.

Q: Does Amazon EMR update the version of Hadoop it supports?

Amazon EMR periodically updates its supported version of Hadoop based on the Hadoop releases by the community. Amazon EMR may choose to skip some Hadoop releases.

Q: How quickly does Amazon EMR retire support for old Hadoop versions?

Amazon EMR service retires support for old Hadoop versions several months after deprecation. However, Amazon EMR APIs are backward compatible, so if you build tools on top of these APIs, they will work even when Amazon EMR updates the Hadoop version it’s using.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Debugging**

Q: How can I debug my cluster?

You first select the cluster you want to debug, then click on the “Debug” button to access the debug a cluster window in the AWS Management Console. This will enable you to track progress and identify issues in steps, jobs, tasks, or task attempts of your clusters. Alternatively you can SSH directly into the Amazon Elastic Compute Cloud (Amazon EC2) instances that are running your cluster and use your favorite command-line debugger to troubleshoot the cluster.

Q: What is the cluster debug tool?

The cluster debug tool is a part of the AWS Management Console where you can track progress and identify issues in steps, jobs, tasks, or task attempts of your clusters. To access the cluster debug tool, first select the cluster you want to debug and then click on the “Debug” button.

Q: How can I enable debugging of my cluster?

To enable debugging you need to set “Enable Debugging” flag when you create a cluster in the AWS Management Console. Alternatively, you can pass the --enable-debugging and --log-uri flags in the Command Line Client when creating a cluster.

Q: Where can I find instructions on how to use the debug a cluster window?

Please reference the AWS Management Console section of the [Developer’s Guide](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/) for instructions on how to access and use the debug a cluster window.

Q: What types of clusters can I debug with the debug a cluster window?

You can debug all types of clusters currently supported by Amazon EMR including custom jar, streaming, Hive, and Pig.

Q: Why do I have to sign-up for Amazon SimpleDB to use cluster debugging?

Amazon EMR stores state information about Hadoop jobs, tasks and task attempts under your account in Amazon SimpleDB. You can subscribe to Amazon SimpleDB [here](https://aws.amazon.com/simpledb/).

Q: Can I use the cluster debugging feature without Amazon SimpleDB subscription?

You will be able to browse cluster steps and step logs but will not be able to browse Hadoop jobs, tasks, or task attempts if you are not subscribed to Amazon SimpleDB.

Q: Can I delete historical cluster data from Amazon SimpleDB?

Yes. You can delete Amazon SimpleDB domains that Amazon EMR created on your behalf. Please reference the [Amazon SimpleDB documentation](http://aws.amazon.com/documentation/simpledb/) for instructions.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Managing Data**

Q: How do I get my data into Amazon S3?

You can use Amazon S3 APIs to upload data to Amazon S3. Alternatively, you can use many open source or commercial [clients](https://d0.awsstatic.com/whitepapers/Storage/An_Overview_of_AWS_Cloud_Data_Migration_Services.pdf) to easily upload data to Amazon S3.

Q: How do I get logs for completed clusters?

Hadoop system logs as well as user logs will be placed in the Amazon S3 bucket which you specify when creating a cluster.

Q: Do you compress logs?

No. At this time Amazon EMR does not compress logs as it moves them to Amazon S3.

Q: Can I load my data from the internet or somewhere other than Amazon S3?

Yes. Your Hadoop application can load the data from anywhere on the internet or from other AWS services. Note that if you load data from the internet, [EC2 bandwidth charges](https://aws.amazon.com/ec2/pricing/) will apply. Amazon EMR also provides Hive-based access to data in [DynamoDB](https://aws.amazon.com/dynamodb/).

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Billing**

Q: Can Amazon EMR estimate how long it will take to process my input data?

No. As each cluster and input data is different, we cannot estimate your job duration.

Q: How much does Amazon EMR cost?

As with the rest of AWS, you pay only for what you use. There is no minimum fee and there are no up-front commitments or long-term contracts. Amazon EMR pricing is in addition to normal Amazon EC2 and Amazon S3 pricing.

For Amazon EMR pricing information, please visit EMR's [pricing page](https://aws.amazon.com/emr/pricing/).

Amazon EC2, Amazon S3 and Amazon SimpleDB charges are billed separately. Pricing for Amazon EMR is per-second consumed for each instance type (with a one-minute minimum), from the time cluster is requested until it is terminated. For additional details on Amazon EC2 Instance Types, Amazon EC2 Spot Pricing, Amazon EC2 Reserved Instances Pricing, Amazon S3 Pricing, or Amazon SimpleDB Pricing, follow the links below:

[Amazon EC2 Instance Types](https://aws.amazon.com/ec2/instance-types/)

[Amazon EC2 Reserved Instances Pricing](https://aws.amazon.com/ec2/pricing/)

[Amazon EC2 Spot Instances Pricing](https://aws.amazon.com/ec2/pricing/)

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

[Amazon SimpleDB Pricing](https://aws.amazon.com/simpledb/pricing/)

Q: When does billing of my Amazon EMR cluster begin and end?

Billing commences when Amazon EMR starts running your cluster. You are only charged for the resources actually consumed. For example, let’s say you launched 100 Amazon EC2 Standard Small instances for an Amazon EMR cluster, where the Amazon EMR cost is an incremental $0.015 per hour. The Amazon EC2 instances will begin booting immediately, but they won’t necessarily all start at the same moment. Amazon EMR will track when each instance starts and will check it into the cluster so that it can accept processing tasks.

In the first 10 minutes after your launch request, Amazon EMR either starts your cluster (if all of your instances are available) or checks in as many instances as possible. Once the 10 minute mark has passed, Amazon EMR will start processing (and charging for) your cluster as soon as 90% of your requested instances are available. As the remaining 10% of your requested instances check in, Amazon EMR starts charging for those instances as well.

So, in the above example, if all 100 of your requested instances are available 10 minutes after you kick off a launch request, you’ll be charged $1.50 per hour (100 \* $0.015) for as long as the cluster takes to complete. If only 90 of your requested instances were available at the 10 minute mark, you’d be charged $1.35 per hour (90 \* $0.015) for as long as this was the number of instances running your cluster. When the remaining 10 instances checked in, you’d be charged $1.50 per hour (100 \* $0.015) for as long as the balance of the cluster takes to complete.

Each cluster will run until one of the following occurs: you terminate the cluster with the TerminateJobFlows API call (or an equivalent tool), the cluster shuts itself down, or the cluster is terminated due to software or hardware failure.

Q: Where can I track my Amazon EMR, Amazon EC2 and Amazon S3 usage?

You can track your usage in the [Billing & Cost Management Console](https://console.aws.amazon.com/billing/home).

Q: How do you calculate the Normalized Instance Hours displayed on the console ?

On the AWS Management Console, every cluster has a Normalized Instance Hours column that displays the approximate number of compute hours the cluster has used, rounded up to the nearest hour. Normalized Instance Hours are hours of compute time based on the standard of 1 hour of m1.small usage = 1 hour normalized compute time. The following table outlines the normalization factor used to calculate normalized instance hours for the various instance sizes:

|  |  |
| --- | --- |
| Instance Size | Normalization Factor |
| Small | 1 |
| Medium | 2 |
| Large | 4 |
| Xlarge | 8 |
| 2xlarge | 16 |
| 4xlarge | 32 |
| 8xlarge | 64 |

For example, if you run a 10-node r3.8xlarge cluster for an hour, the total number of Normalized Instance Hours displayed on the console will be 640 (10 (number of nodes) x 64 (normalization factor) x 1 (number of hours tthat the cluster ran) = 640).

This is an approximate number and should not be used for billing purposes. Please refer to the [Billing & Cost Management Console](https://console.aws.amazon.com/billing/home#/) for billable Amazon EMR usage. Note that we recently changed the normalization factor to accurately reflect the weights of the instances, and the normalization factor does not affect your monthly bill.

Q: Does Amazon EMR support Amazon EC2 On-Demand, Spot, and Reserved Instances?

Yes. Amazon EMR seamlessly supports On-Demand, Spot, and Reserved Instances. [Click here](http://aws.amazon.com/ec2/faqs/#What_is_a_Reserved_Instance) to learn more about Amazon EC2 Reserved Instances. [Click here](http://aws.amazon.com/ec2/spot/) to learn more about Amazon EC2 Spot Instances.

Q: Do your prices include taxes?

Except as otherwise noted, our prices are exclusive of applicable taxes and duties, including VAT and applicable sales tax. For customers with a Japanese billing address, use of AWS services is subject to Japanese Consumption Tax. [Learn more.](http://aws.amazon.com/c-tax-faqs/)

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Security**

Q: How do I prevent other people from viewing my data during cluster execution?

Amazon EMR starts your instances in two Amazon EC2 security groups, one for the master and another for the slaves. The master security group has a port open for communication with the service. It also has the SSH port open to allow you to SSH into the instances, using the key specified at startup. The slaves start in a separate security group, which only allows interaction with the master instance. By default both security groups are set up to not allow access from external sources including Amazon EC2 instances belonging to other customers. Since these are security groups within your account, you can reconfigure them using the standard EC2 tools or dashboard. [Click here](http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/using-network-security.html) to learn more about EC2 security groups.

Q: How secure is my data?

Amazon S3 provides authentication mechanisms to ensure that stored data is secured against unauthorized access. Unless the customer who is uploading the data specifies otherwise, only that customer can access the data. Amazon EMR customers can also choose to send data to Amazon S3 using the HTTPS protocol for secure transmission. In addition, Amazon EMR always uses HTTPS to send data between Amazon S3 and Amazon EC2. For added security, customers may encrypt the input data before they upload it to Amazon S3 (using any common data encryption tool); they then need to add a decryption step to the beginning of their cluster when Amazon EMR fetches the data from Amazon S3.

Q: Can I get a history of all EMR API calls made on my account for security or compliance auditing?

Yes. AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing. Learn more about CloudTrail at the [AWS CloudTrail detail page](http://aws.amazon.com/cloudtrail/), and turn it on via [CloudTrail's AWS Management Console](https://console.aws.amazon.com/cloudtrail).

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Regions & Availability Zones**

Q: How does Amazon EMR make use of Availability Zones?

Amazon EMR launches all nodes for a given cluster in the same Amazon EC2 Availability Zone. Running a cluster in the same zone improves performance of the jobs flows because it provides a higher data access rate. By default, Amazon EMR chooses the Availability Zone with the most available resources in which to run your cluster. However, you can specify another Availability Zone if required.

Q: In what Regions is this Amazon EMR available?

For a list of the supported Amazon EMR AWS regions, please visit the [AWS Region Table](https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/) for all AWS global infrastructure.

Q: Which Region should I select to run my clusters?

When creating a cluster, typically you should select the Region where your data is located.

Q: Can I use EU data in a cluster running in the US region and vice versa?

Yes you can. If you transfer data from one region to the other you will be charged bandwidth charges. For bandwidth pricing information, please visit the pricing section on the [EC2 detail page](https://aws.amazon.com/ec2/).

Q: What is different about the AWS GovCloud (US) region?

The [AWS GovCloud (US)](https://aws.amazon.com/govcloud-us/) region is designed for US government agencies and customers. It adheres to US ITAR requirements. In GovCloud, EMR does not support spot instances or the enable-debugging feature. The EMR Management Console is not yet available in GovCloud.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Managing your Cluster**

Q: How does Amazon EMR use Amazon EC2 and Amazon S3?

Customers upload their input data and a data processing application into Amazon S3. Amazon EMR then launches a number of Amazon EC2 instances as specified by the customer. The service begins the cluster execution while pulling the input data from Amazon S3 using S3N protocol into the launched Amazon EC2 instances. Once the cluster is finished, Amazon EMR transfers the output data to Amazon S3, where customers can then retrieve it or use as input in another cluster.

Q: How is a computation done in Amazon EMR?

Amazon EMR uses the Hadoop data processing engine to conduct computations implemented in the MapReduce programming model. The customer implements their algorithm in terms of map() and reduce() functions. The service starts a customer-specified number of Amazon EC2 instances, comprised of one master and multiple slaves. Amazon EMR runs Hadoop software on these instances. The master node divides input data into blocks, and distributes the processing of the blocks to the slave node. Each slave node then runs the map function on the data it has been allocated, generating intermediate data. The intermediate data is then sorted and partitioned and sent to processes which apply the reducer function to it. These processes also run on the slave nodes. Finally, the output from the reducer tasks is collected in files. A single “cluster” may involve a sequence of such MapReduce steps.

Q: How reliable is Amazon EMR?

Amazon EMR manages an Amazon EC2 cluster of compute instances using Amazon’s highly available, proven network infrastructure and datacenters. Amazon EMR uses industry proven, fault-tolerant Hadoop software as its data processing engine. Hadoop splits the data into multiple subsets and assigns each subset to more than one Amazon EC2 instance. So, if an Amazon EC2 instance fails to process one subset of data, the results of another Amazon EC2 instance can be used.

Q: How quickly will my cluster be up and running and processing my input data?

Amazon EMR starts resource provisioning of Amazon EC2 On-Demand instances almost immediately. If the instances are not available, Amazon EMR will keep trying to provision the resources for your cluster until they are provisioned or you cancel your request. The instance provisioning is done on a best-efforts basis and depends on the number of instances requested, time when the cluster is created, and total number of requests in the system. After resources have been provisioned, it typically takes fewer than 15 minutes to start processing.

In order to guarantee capacity for your clusters at the time you need it, you may pay a one-time fee for Amazon EC2 Reserved Instances to reserve instance capacity in [the cloud](https://aws.amazon.com/what-is-cloud-computing/) at a discounted hourly rate. Like On-Demand Instances, customers pay usage charges only for the time when their instances are running. In this way, Reserved Instances enable businesses with known instance requirements to maintain the elasticity and flexibility of On-Demand Instances, while also reducing their predictable usage costs even further.

Q: Which Amazon EC2 instance types does Amazon EMR support?

Amazon EMR supports 12 EC2 instance types including Standard, High CPU, High Memory, Cluster Compute, High I/O, and High Storage. Standard Instances have memory to CPU ratios suitable for most general-purpose applications. High CPU instances have proportionally more CPU resources than memory (RAM) and are well suited for compute-intensive applications. High Memory instances offer large memory sizes for high throughput applications. Cluster Compute instances have proportionally high CPU with increased network performance and are well suited for High Performance Compute (HPC) applications and other demanding network-bound applications. High Storage instances offer 48 TB of storage across 24 disks and are ideal for applications that require sequential access to very large data sets such as data warehousing and log processing. See the [EMR pricing page](https://aws.amazon.com/emr/pricing/) for details on available instance types and pricing per region.

Q: How do I select the right Amazon EC2 instance type?

When choosing instance types, you should consider the characteristics of your application with regards to resource utilization and select the optimal instance family. One of the advantages of Amazon EMR with Amazon EC2 is that you pay only for what you use, which makes it convenient and inexpensive to test the performance of your clusters on different instance types and quantity. One effective way to determine the most appropriate instance type is to launch several small clusters and benchmark your clusters.

Q: How do I select the right number of instances for my cluster?

The number of instances to use in your cluster is application-dependent and should be based on both the amount of resources required to store and process your data and the acceptable amount of time for your job to complete. As a general guideline, we recommend that you limit 60% of your disk space to storing the data you will be processing, leaving the rest for intermediate output. Hence, given 3x replication on HDFS, if you were looking to process 5 TB on m1.xlarge instances, which have 1,690 GB of disk space, we recommend your cluster contains at least (5 TB \* 3) / (1,690 GB \* .6) = 15 m1.xlarge core nodes. You may want to increase this number if your job generates a high amount of intermediate data or has significant I/O requirements. You may also want to include additional task nodes to improve processing performance. See [Amazon EC2 Instance Types](https://aws.amazon.com/ec2/instance-types/) for details on local instance storage for each instance type configuration.

Q: How long will it take to run my cluster?

The time to run your cluster will depend on several factors including the type of your cluster, the amount of input data, and the number and type of Amazon EC2 instances you choose for your cluster.

Q: If the master node in a cluster goes down, can Amazon EMR recover it?

No. If the master node goes down, your cluster will be terminated and you’ll have to rerun your job. Amazon EMR currently does not support automatic failover of the master nodes or master node state recovery. In case of master node failure, the AWS Management console displays “The master node was terminated” message which is an indicator for you to start a new cluster. Customers can instrument check pointing in their clusters to save intermediate data (data created in the middle of a cluster that has not yet been reduced) on Amazon S3. This will allow resuming the cluster from the last check point in case of failure.

Q: If a slave node goes down in a cluster, can Amazon EMR recover from it?

Yes. Amazon EMR is fault tolerant for slave failures and continues job execution if a slave node goes down. Amazon EMR will also provision a new node when a core node fails. However, Amazon EMR will not replace nodes if all nodes in the cluster are lost.

Q: Can I SSH onto my cluster nodes?

Yes. You can SSH onto your cluster nodes and execute Hadoop commands directly from there. If you need to SSH into a slave node, you have to first SSH to the master node, and then SSH into the slave node.

Q: What is Amazon EMR Bootstrap Actions?

Bootstrap Actions is a feature in Amazon EMR that provides users a way to run custom set-up prior to the execution of their cluster. Bootstrap Actions can be used to install software or configure instances before running your cluster. You can read more about bootstrap actions in EMR's [Developer Guide.](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-plan-bootstrap.html)

Q: How can I use Bootstrap Actions?

You can write a Bootstrap Action script in any language already installed on the cluster instance including Bash, Perl, Python, Ruby, C++, or Java. There are several pre-defined Bootstrap Actions available. Once the script is written, you need to upload it to Amazon S3 and reference its location when you start a cluster. Please refer to the “Developer’s Guide”: http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/ for details on how to use Bootstrap Actions.

Q: How do I configure Hadoop settings for my cluster?

The EMR default Hadoop configuration is appropriate for most workloads. However, based on your cluster’s specific memory and processing requirements, it may be appropriate to tune these settings. For example, if your cluster tasks are memory-intensive, you may choose to use fewer tasks per core and reduce your job tracker heap size. For this situation, a pre-defined Bootstrap Action is available to configure your cluster on startup. See the [Configure Memory Intensive Bootstrap Action](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/Bootstrap.html#PredefinedBootstrapActions_MemoryIntensive) in the Developer’s Guide for configuration details and usage instructions. An additional predefined bootstrap action is available that allows you to customize your cluster settings to any value of your choice. See the [Configure Hadoop Bootstrap Action](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/Bootstrap.html#PredefinedBootstrapActions_ConfigureHadoop) in the Developer’s Guide for usage instructions.

Q: Can I modify the number of slave nodes in a running cluster?

Yes. Slave nodes can be of two types: (1) core nodes, which both host persistent data using Hadoop Distributed File System (HDFS) and run Hadoop tasks and (2) task nodes, which only run Hadoop tasks. While a cluster is running you may increase the number of core nodes and you may either increase or decrease the number of task nodes. This can be done through the API, Java SDK, or though the command line client. Please refer to the [Resizing Running clusters](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/ResizeJobFlow.html) section in the Developer’s Guide for details on how to modify the size of your running cluster.

Q: When would I want to use core nodes versus task nodes?

As core nodes host persistent data in HDFS and cannot be removed, core nodes should be reserved for the capacity that is required until your cluster completes. As task nodes can be added or removed and do not contain HDFS, they are ideal for capacity that is only needed on a temporary basis.

Q: Why would I want to modify the number of slave nodes in my running cluster?

There are several scenarios where you may want to modify the number of slave nodes in a running cluster. If your cluster is running slower than expected, or timing requirements change, you can increase the number of core nodes to increase cluster performance. If different phases of your cluster have different capacity needs, you can start with a small number of core nodes and increase or decrease the number of task nodes to meet your cluster’s varying capacity requirements.

Q: Can I automatically modify the number of slave nodes between cluster steps?

Yes. You may include a predefined step in your workflow that automatically resizes a cluster between steps that are known to have different capacity needs. As all steps are guaranteed to run sequentially, this allows you to set the number of slave nodes that will execute a given cluster step.

Q: How can I allow other IAM users to access my cluster?

To create a new cluster that is visible to all IAM users within the EMR CLI: Add the --visible-to-all-users flag when you create the cluster. For example: elastic-mapreduce --create --visible-to-all-users. Within the Management Console, simply select “Visible to all IAM Users” on the Advanced Options pane of the Create cluster Wizard.

To make an existing cluster visible to all IAM users you must use the EMR CLI. Use --set-visible-to-all-users and specify the cluster identifier. For example: elastic-mapreduce --set-visible-to-all-users true --jobflow j-xxxxxxx. This can only be done by the creator of the cluster.

To learn more, see the [Configuring User Permissions](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/environmentconfig_iam.html) section of the EMR Developer Guide.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Tagging your Cluster**

Q: What Amazon EMR resources can I tag?

You can add tags to an active Amazon EMR cluster. An Amazon EMR cluster consists of Amazon EC2 instances, and a tag added to an Amazon EMR cluster will be propagated to each active Amazon EC2 instance in that cluster. You cannot add, edit, or remove tags from terminated clusters or terminated Amazon EC2 instances which were part of an active cluster.

Q: Does Amazon EMR tagging support resource-based permissions with IAM Users?

No, Amazon EMR does not support resource-based permissions by tag. However, it is important to note that propagated tags to Amazon EC2 instances behave as normal Amazon EC2 tags. Therefore, an IAM Policy for Amazon EC2 will act on tags propagated from Amazon EMR if they match conditions in that policy.

Q: How many tags can I add to a resource?

You can add up to ten tags on an Amazon EMR cluster.

Q: Do my Amazon EMR tags on a cluster show up on each Amazon EC2 instance in that cluster? If I remove a tag on my Amazon EMR cluster, will that tag automatically be removed from each associated EC2 instance?

Yes, Amazon EMR propagates the tags added to a cluster to that cluster's underlying EC2 instances. If you add a tag to an Amazon EMR cluster, it will also appear on the related Amazon EC2 instances. Likewise, if you remove a tag from an Amazon EMR cluster, it will also be removed from its associated Amazon EC2 instances. However, if you are using IAM policies for Amazon EC2 and plan to use Amazon EMR's tagging functionality, you should make sure that permission to use the Amazon EC2 tagging APIs CreateTags and DeleteTags is granted.

Q: How do I get my tags to show up in my billing statement to segment costs?

Select the tags you would like to use in your AWS billing report [here](https://console.aws.amazon.com/billing/home?#/preferences/tags). Then, to see the cost of your combined resources, you can organize your billing information based on resources that have the same tag key values.

Q: How do I tell which Amazon EC2 instances are part of an Amazon EMR cluster?

An Amazon EC2 instance associated with an Amazon EMR cluster will have two system tags:

* aws:elasticmapreduce:instance-group-role=CORE
  + Key = instance-group role ; Value = [CORE or TASK]
* aws:elasticmapreduce:job-flow-id=j-12345678
  + Key = job-flow-id ; Value = [JobFlowID]

Q: Can I edit tags directly on the Amazon EC2 instances?

Yes, you can add or remove tags directly on Amazon EC2 instances that are part of an Amazon EMR cluster. However, we do not recommend doing this, because Amazon EMR’s tagging system will not sync the changes you make to an associated Amazon EC2 instance directly. We recommend that tags for Amazon EMR clusters be added and removed from the Amazon EMR console, CLI, or API to ensure that the cluster and its associated Amazon EC2 instances have the correct tags.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Using EBS Volumes**

Q: What can I do now that I could not do before?

Most EC2 instances have fixed storage capacity attached to an instance, known as an "instance store". You can now add EBS volumes to the instances in your Amazon EMR cluster, allowing you to customize the storage on an instance. The feature also allows you to run Amazon EMR clusters on EBS-Only instance families such as the M4 and C4.

Q: What are the benefits of adding EBS volumes to an instance running on Amazon EMR?

You will benefit by adding EBS volumes to an instance in the following scenarios:

1. Your processing requirements are such that you need a large amount of HDFS (or local) storage that what is available today on an instance. With support for EBS volumes, you will be able to customize the storage capacity on an instance relative to the compute capacity that the instance provides. Optimizing the storage on an instance will allow you to save costs.
2. You are running on an older generation instance family (such as the M1 and M2 family) and want to move to latest generation instance family but are constrained by the storage available per node on the next generation instance types. Now you can use any of the new generation instance type and add EBS volumes to optimize the storage.  Internal benchmarks indicate that you can save cost and improve performance by moving from an older generation instance family (M1 or M2) to a new generation one (M4, C4 & R3).  The Amazon EMR team recommends that you run your application to arrive at the right conclusion.
3. You want to use or migrate to the next-generation EBS-Only M4 and C4 family.

Q: Can I persist my data on an EBS volume after a cluster is terminated?

Currently, Amazon EMR will delete volumes once the cluster is terminated. If you want to persist data outside the lifecycle of a cluster, consider using Amazon S3 as your data store.

Q: What kind of EBS volumes can I attach to an instance?

Amazon EMR allows you to use different EBS Volume Types: General Purpose SSD (GP2), Magnetic and Provisioned IOPS (SSD).

Q: What happens to the EBS volumes once I terminate my cluster?

Amazon EMR will delete the volumes once the EMR cluster is terminated.

Q: Can I use an EBS with instances that already have an instance store?

Yes, You can add EBS volumes to instances that have an instance store.

Q: Can I attach an EBS volume to a running cluster?

No, currently you can only add EBS volumes when launching a cluster.

Q: Can I snapshot volumes from a cluster?

The EBS API allows you to Snapshot a cluster. However, Amazon EMR currently does not allow you to restore from a snapshot.

Q: Can I use encrypted EBS volumes?

No, encrypted volumes are not supported in the current release.

Q: What happens when I remove an attached volume from a running cluster?

Removing an attached volume from a running cluster will be treated as a node failure.  Amazon EMR will replace the node and the EBS volume with each of the same.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Using Hive**

Q: What is Apache Hive?

Hive is an open source datawarehouse and analytics package that runs on top of Hadoop. Hive is operated by a SQL-based language called Hive QL that allows users to structure, summarize, and query data sources stored in Amazon S3. Hive QL goes beyond standard SQL, adding first-class support for map/reduce functions and complex extensible user-defined data types like Json and Thrift. This capability allows processing of complex and even unstructured data sources such as text documents and log files. Hive allows user extensions via user-defined functions written in Java and deployed via storage in Amazon S3.

Q: What can I do with Hive running on Amazon EMR?

Using Hive with Amazon EMR, you can implement sophisticated data-processing applications with a familiar SQL-like language and easy to use tools available with Amazon EMR. With Amazon EMR, you can turn your Hive applications into a reliable data warehouse to execute tasks such as data analytics, monitoring, and business intelligence tasks.

Q: How is Hive different than traditional RDBMS systems?

Traditional RDBMS systems provide transaction semantics and ACID properties. They also allow tables to be indexed and cached so that small amounts of data can be retrieved very quickly. They provide for fast update of small amounts of data and for enforcement of referential integrity constraints. Typically they run on a single large machine and do not provide support for executing map and reduce functions on the table, nor do they typically support acting over complex user defined data types.

In contrast, Hive executes SQL-like queries using MapReduce. Consequently, it is optimized for doing full table scans while running on a cluster of machines and is therefore able to process very large amounts of data. Hive provides partitioned tables, which allow it to scan a partition of a table rather than the whole table if that is appropriate for the query it is executing.

Traditional RDMS systems are best for when transactional semantics and referential integrity are required and frequent small updates are performed. Hive is best for offline reporting, transformation, and analysis of large data sets; for example, performing click stream analysis of a large website or collection of websites.

One of the common practices is to export data from RDBMS systems into Amazon S3 where offline analysis can be performed using Amazon EMR clusters running Hive.

Q: How can I get started with Hive running on Amazon EMR?

The best place to start is to review our written documentation located [here](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive.html).

Q: Are there new features in Hive specific to Amazon EMR?

Yes. There are four new features which make Hive even more powerful when used with Amazon EMR, including:

a/ The ability to load table partitions automatically from Amazon S3. Previously, to import a partitioned table you needed a separate alter table statement for each individual partition in the table. Amazon EMR a now includes a new statement type for the Hive language: “alter table recover partitions.” This statement allows you to easily import tables concurrently into many clusters without having to maintain a shared meta-data store. Use this functionality to read from tables into which external processes are depositing data, for example log files.

b/ The ability to specify an off-instance metadata store. By default, the metadata store where Hive stores its schema information is located on the master node and ceases to exist when the cluster terminates. This feature allows you to override the location of the metadata store to use, for example a MySQL instance that you already have running in EC2.

c/ Writing data directly to Amazon S3. When writing data to tables in Amazon S3, the version of Hive installed in Amazon EMR writes directly to Amazon S3 without the use of temporary files. This produces a significant performance improvement but it means that HDFS and S3 from a Hive perspective behave differently. You cannot read and write within the same statement to the same table if that table is located in Amazon S3. If you want to update a table located in S3, then create a temporary table in the cluster’s local HDFS filesystem, write the results to that table, and then copy them to Amazon S3.

d/ Accessing resources located in Amazon S3. The version of Hive installed in Amazon EMR allows you to reference resources such as scripts for custom map and reduce operations or additional libraries located in Amazon S3 directly from within your Hive script (e.g., add jar s3://elasticmapreduce/samples/hive-ads/libs/jsonserde.jar).

Q: What types of Hive clusters are supported?

There are two types of clusters supported with Hive: interactive and batch. In an interactive mode a customer can start a cluster and run Hive scripts interactively directly on the master node. Typically, this mode is used to do ad hoc data analyses and for application development. In batch mode, the Hive script is stored in Amazon S3 and is referenced at the start of the cluster. Typically, batch mode is used for repeatable runs such as report generation.

Q: How can I launch a Hive cluster?

Both batch and interactive clusters can be started from AWS Management Console, EMR command line client, or APIs. Please refer to the [Hive](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-hive.html) section in the Release Guide for more details on launching a Hive cluster.

Q: When should I use Hive vs. PIG?

Hive and PIG both provide high level data-processing languages with support for complex data types for operating on large datasets. The Hive language is a variant of SQL and so is more accessible to people already familiar with SQL and relational databases. Hive has support for partitioned tables which allow Amazon EMR clusters to pull down only the table partition relevant to the query being executed rather than doing a full table scan. Both PIG and Hive have query plan optimization. PIG is able to optimize across an entire scripts while Hive queries are optimized at the statement level.

Ultimately the choice of whether to use Hive or PIG will depend on the exact requirements of the application domain and the preferences of the implementers and those writing queries.

Q: What version of Hive does Amazon EMR support?

Amazon EMR supports multiple versions of Hive, including version 0.11.0.

Q: Can I write to a table from two clusters concurrently

No. Hive does not support concurrently writing to tables. You should avoid concurrently writing to the same table or reading from a table while you are writing to it. Hive has non-deterministic behavior when reading and writing at the same time or writing and writing at the same time.

Q: Can I share data between clusters?

Yes. You can read data in Amazon S3 within a Hive script by having ‘create external table’ statements at the top of your script. You need a create table statement for each external resource that you access.

Q: Should I run one large cluster, and share it amongst many users or many smaller clusters?

Amazon EMR provides a unique capability for you to use both methods. On the one hand one large cluster may be more efficient for processing regular batch workloads. On the other hand, if you require ad-hoc querying or workloads that vary with time, you may choose to create several separate cluster tuned to the specific task sharing data sources stored in Amazon S3.

Q: Can I access a script or jar resource which is on my local file system?

No. You must upload the script or jar to Amazon S3 or to the cluster’s master node before it can be referenced. For uploading to Amazon S3 you can use tools including s3cmd, jets3t or S3Organizer.

Q: Can I run a persistent cluster executing multiple Hive queries?

Yes. You run a cluster in a manual termination mode so it will not terminate between Hive steps. To reduce the risk of data loss we recommend periodically persisting all of your important data in Amazon S3. It is good practice to regularly transfer your work to a new cluster to test your process for recovering from master node failure.

Q: Can multiple users execute Hive steps on the same source data?

Yes. Hive scripts executed by multiple users on separate clusters may contain create external table statements to concurrently import source data residing in Amazon S3.

Q: Can multiple users run queries on the same cluster?

Yes. In the batch mode, steps are serialized. Multiple users can add Hive steps to the same cluster, however, the steps will be executed serially. In interactive mode, several users can be logged on to the same cluster and execute Hive statements concurrently.

Q: Can data be shared between multiple AWS users?

Yes. Data can be shared using standard Amazon S3 sharing mechanism described [here](http://docs.amazonwebservices.com/AmazonS3/latest/index.html?S3_ACLs.html).

Q: Does Hive support access from JDBC?

Yes. Hive provides JDBC drive, which can be used to programmatically execute Hive statements. To start a JDBC service in your cluster you need to pass an optional parameter in the Amazon EMR command line client. You also need to establish an SSH tunnel because the security group does not permit external connections.

Q: What is your procedure for updating packages on EMR AMIs?

On first boot, the Amazon Linux AMIs for EMR connect to the Amazon Linux AMI yum repositories to install security updates. When you use a custom AMI, you can disable this feature, but we don’t recommend this for security reasons.

Q: Can I update my own packages on EMR clusters?

Yes. You can use Bootstrap Actions to install updates to packages on your clusters.

Q: Can I process DynamoDB data using Hive?

Yes. Simply define an external Hive table based on your DynamoDB table. You can then use Hive to analyze the data stored in DynamoDB and either load the results back into DynamoDB or archive them in Amazon S3. For more information please visit our [Developer Guide](http://docs.amazonwebservices.com/ElasticMapReduce/latest/DeveloperGuide/EMRforDynamoDB.html).

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Using Impala**

Q: What is Impala?

Impala is an open source tool in the Hadoop ecosystem for interactive, ad hoc querying using SQL syntax. Instead of using MapReduce, it leverages a massively parallel processing (MPP) engine similar to that found in traditional relational database management systems (RDBMS). With this architecture, you can query your data in HDFS or HBase tables very quickly, and leverage Hadoop’s ability to process diverse data types and provide schema at runtime. This lends Impala to interactive, low-latency analytics. In addition, Impala uses the Hive metastore to hold information about the input data, including the partition names and data types. Also, Impala on Amazon EMR requires AMIs running Hadoop 2.x or greater. Click [here](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-impala.html) to learn more about Impala.

Q: What can I do with Impala running on Amazon EMR?

Similar to using Hive with Amazon EMR, leveraging Impala with Amazon EMR can implement sophisticated data-processing applications with SQL syntax. However, Impala is built to perform faster in certain use cases (see below). With Amazon EMR, you can use Impala as a reliable data warehouse to execute tasks such as data analytics, monitoring, and business intelligence. Here are three use cases:

* Use Impala instead of Hive on long-running clusters to perform ad hoc queries. Impala reduces interactive queries to seconds, making it an excellent tool for fast investigation. You could run Impala on the same cluster as your batch MapReduce workflows, use Impala on a long-running analytics cluster with Hive and Pig, or create a cluster specifically tuned for Impala queries.
* Use Impala instead of Hive for batch ETL jobs on transient Amazon EMR clusters. Impala is faster than Hive for many queries, which provides better performance for these workloads. Like Hive, Impala uses SQL, so queries can easily be modified from Hive to Impala.
* Use Impala in conjunction with a third party business intelligence tool. Connect a client ODBC or JDBC driver with your cluster to use Impala as an engine for powerful visualization tools and dashboards.

Both batch and interactive Impala clusters can be created in Amazon EMR. For instance, you can have a long-running Amazon EMR cluster running Impala for ad hoc, interactive querying or use transient Impala clusters for quick ETL workflows.

Q: How is Impala different than traditional RDBMSs?

Traditional relational database systems provide transaction semantics and database atomicity, consistency, isolation, and durability (ACID) properties. They also allow tables to be indexed and cached so that small amounts of data can be retrieved very quickly, provide for fast updates of small amounts of data, and for enforcement of referential integrity constraints. Typically, they run on a single large machine and do not provide support for acting over complex user defined data types. Impala uses a similar distributed query system to that found in RDBMSs, but queries data stored in HDFS and uses the Hive metastore to hold information about the input data. As with Hive, the schema for a query is provided at runtime, allowing for easier schema changes. Also, Impala can query a variety of complex data types and execute user defined functions. However, because Impala processes data in-memory, it is important to understand the hardware limitations of your cluster and optimize your queries for the best performance.

Q: How is Impala different than Hive?

Impala executes SQL queries using a massively parallel processing (MPP) engine, while Hive executes SQL queries using MapReduce. Impala avoids Hive’s overhead from creating MapReduce jobs, giving it faster query times than Hive. However, Impala uses significant memory resources and the cluster’s available memory places a constraint on how much memory any query can consume. Hive is not limited in the same way, and can successfully process larger data sets with the same hardware. Generally, you should use Impala for fast, interactive queries, while Hive is better for ETL workloads on large datasets. Impala is built for speed and is great for ad hoc investigation, but requires a significant amount of memory to execute expensive queries or process very large datasets. Because of these limitations, Hive is recommended for workloads where speed is not as crucial as completion. Click [here](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/impala-optimization.html) to view some performance benchmarks between Impala and Hive.

Q: Can I use Hadoop 1?

No, Impala requires Hadoop 2, and will not run on a cluster with an AMI running Hadoop 1.x.

Q: What instance types should I use for my Impala cluster?

For the best experience with Impala, we recommend using memory-optimized instances for your cluster. However, we have shown that there are performance gains over Hive when using standard instance types as well. We suggest reading our [Performance Testing and Query Optimization](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/impala-optimization.html) section in the Amazon EMR Developer’s Guide to better estimate the memory resources your cluster will need with regards to your dataset and query types. The compression type, partitions, and the actual query (number of joins, result size, etc.) all play a role in the memory required. You can use the EXPLAIN statement to estimate the memory and other resources needed for an Impala query.

Q: What happens if I run out of memory on a query?

If you run out of memory, queries fail and the Impala daemon installed on the affected node shuts down. Amazon EMR then restarts the daemon on that node so that Impala will be ready to run another query. Your data in HDFS on the node remains available, because only the daemon running on the node shuts down, rather than the entire node itself. For ad hoc analysis with Impala, the query time can often be measured in seconds; therefore, if a query fails, you can discover the problem quickly and be able to submit a new query in quick succession.

Q: Does Impala support user defined functions?

Yes, Impala supports user defined functions (UDFs). You can write Impala specific UDFs in Java or C++. Also, you can modify UDFs or user-defined aggregate functions created for Hive for use with Impala. For information about Hive UDFs, click [here](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF).

Q: Where is the data stored for Impala to query?

Impala queries data in HDFS or in HBase tables.

Q: Can I run Impala and MapReduce at the same time on a cluster?

Yes, you can set up a multitenant cluster with Impala and MapReduce. However, you should be sure to allot resources (memory, disk, and CPU) to each application using YARN on Hadoop 2.x. The resources allocated should be dependent on the needs for the jobs you plan to run on each application.

Q: Does Impala support ODBC and JDBC drivers?

While you can use ODBC drivers, Impala is also a great engine for third-party tools connected through JDBC. You can download and install the Impala client JDBC driver from http://elasticmapreduce.s3.amazonaws.com/libs/impala/1.2.1/impala-jdbc-1.2.1.zip. From the client computer where you have your business intelligence tool installed, connect the JDBC driver to the master node of an Impala cluster using SSH or a VPN on port 21050. For more information, see [Open an SSH Tunnel to the Master Node](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-ssh-tunnel.html).

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Using Pig**

Q: What is Apache Pig?

Pig is an open source analytics package that runs on top of Hadoop. Pig is operated by a SQL-like language called Pig Latin, which allows users to structure, summarize, and query data sources stored in Amazon S3. As well as SQL-like operations, Pig Latin also adds first-class support for map/reduce functions and complex extensible user defined data types. This capability allows processing of complex and even unstructured data sources such as text documents and log files. Pig allows user extensions via user-defined functions written in Java and deployed via storage in Amazon S3.

Q: What can I do with Pig running on Amazon EMR?

Using Pig with Amazon EMR, you can implement sophisticated data-processing applications with a familiar SQL-like language and easy to use tools available with Amazon EMR. With Amazon EMR, you can turn your Pig applications into a reliable data warehouse to execute tasks such as data analytics, monitoring, and business intelligence tasks.

Q: How can I get started with Pig running on Amazon EMR?

The best place to start is to review our written documentation located [here](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-pig.html).

Q: Are there new features in Pig specific to Amazon EMR?

Yes. There are three new features which make Pig even more powerful when used with Amazon EMR, including:

a/ Accessing multiple filesystems. By default a Pig job can only access one remote file system, be it an HDFS store or S3 bucket, for input, output and temporary data. EMR has extended Pig so that any job can access as many file systems as it wishes. An advantage of this is that temporary intra-job data is always stored on the local HDFS, leading to improved perfomance.

b/ Loading resources from S3. EMR has extended Pig so that custom JARs and scripts can come from the S3 file system, for example “REGISTER s3:///my-bucket/piggybank.jar”

c/ Additional Piggybank function for String and DateTime processing.

Q: What types of Pig clusters are supported?

There are two types of clusters supported with Pig: interactive and batch. In an interactive mode a customer can start a cluster and run Pig scripts interactively directly on the master node. Typically, this mode is used to do ad hoc data analyses and for application development. In batch mode, the Pig script is stored in Amazon S3 and is referenced at the start of the cluster. Typically, batch mode is used for repeatable runs such as report generation.

Q: How can I launch a Pig cluster?

Both batch and interactive clusters can be started from AWS Management Console, EMR command line client, or APIs.

Q: What version of Pig does Amazon EMR support?

Amazon EMR supports [multiple versions](https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-pig.html) of Pig.

Q: Can I write to a S3 bucket from two clusters concurrently

Yes, you are able to write to the same bucket from two concurrent clusters.

Q: Can I share input data in S3 between clusters?

Yes, you are able to read the same data in S3 from two concurrent clusters.

Q: Can data be shared between multiple AWS users?

Yes. Data can be shared using standard Amazon S3 sharing mechanism described here <http://docs.amazonwebservices.com/AmazonS3/latest/index.html?S3_ACLs.html>

Q: Should I run one large cluster, and share it amongst many users or many smaller clusters?

Amazon EMR provides a unique capability for you to use both methods. On the one hand one large cluster may be more efficient for processing regular batch workloads. On the other hand, if you require ad-hoc querying or workloads that vary with time, you may choose to create several separate cluster tuned to the specific task sharing data sources stored in Amazon S3.

Q: Can I access a script or jar resource which is on my local file system?

No. You must upload the script or jar to Amazon S3 or to the cluster’s master node before it can be referenced. For uploading to Amazon S3 you can use tools including s3cmd, jets3t or S3Organizer.

Q: Can I run a persistent cluster executing multiple Pig queries?

Yes. You run a cluster in a manual termination mode so it will not terminate between Pig steps. To reduce the risk of data loss we recommend periodically persisting all important data in Amazon S3. It is good practice to regularly transfer your work to a new cluster to test you process for recovering from master node failure.

Q: Does Pig support access from JDBC?

No. Pig does not support access through JDBC.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Using HBase**

Q: What is Apache HBase?

HBase is an open source, non-relational, distributed database modeled after Google's BigTable. It was developed as part of Apache Software Foundation's Hadoop project and runs on top of Hadoop Distributed File System(HDFS) to provide BigTable-like capabilities for Hadoop. HBase provides you a fault-tolerant, efficient way of storing large quantities of sparse data using column-based compression and storage. In addition, HBase provides fast lookup of data because data is stored in-memory instead of on disk. HBase is optimized for sequential write operations, and it is highly efficient for batch inserts, updates, and deletes. HBase works seamlessly with Hadoop, sharing its file system and serving as a direct input and output to Hadoop jobs. HBase also integrates with Apache Hive, enabling SQL-like queries over HBase tables, joins with Hive-based tables, and support for Java Database Connectivity (JDBC).

Q: Are there new features in HBase specific to Amazon EMR?

With Amazon EMR you can back up HBase to Amazon S3 (full or incremental, manual or automated) and you can restore from a previously created backup. [Learn more about HBase and EMR](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-hbase.html).

Q: Which versions of HBase are supported on Amazon EMR?

Amazon EMR supports HBase 0.94.7 and HBase 0.92.0. To use HBase 0.94.7 you must specify AMI version 3.0.0. If you are using the CLI you must use version 2013-10-07 or later.

[Return to Top >>](https://aws.amazon.com/emr/faqs/#faqs)

## **Kinesis Connector**

Q: What does EMR Connector to Kinesis enable?

The connector enables EMR to directly read and query data from Kinesis streams. You can now perform batch processing of Kinesis streams using existing Hadoop ecosystem tools such as Hive, Pig, MapReduce, Hadoop Streaming, and Cascading.

Q: What does the EMR connector to Kinesis enable that I couldn’t have done before?

Reading and processing data from a Kinesis stream would require you to write, deploy and maintain independent stream processing applications. These take time and effort. However, with this connector, you can start reading and analyzing a Kinesis stream by writing a simple Hive or Pig script. This means you can analyze Kinesis streams using SQL! Of course, other Hadoop ecosystem tools could be used as well. You don’t need to developed or maintain a new set of processing applications.

Q: Who will find this functionality useful?

The following types of users will find this integration useful:

* Hadoop users who are interested in utilizing the extensive set of Hadoop ecosystem tools to analyze Kinesis streams.
* Kinesis users who are looking for an easy way to get up and running with stream processing and ETL of Kinesis data.
* Business analysts and IT professionals who would like to perform ad-hoc analysis of data in Kinesis streams using familiar tools like SQL (via Hive) or scripting languages like Pig.

Q: What are some use cases for this integration?

The following are representative use cases are enabled by this integration:

* Streaming Log Analysis: You can analyze streaming web logs to generate a list of top 10 error type every few minutes by region, browser, and access domains.
* Complex Data Processing Workflows: You can join Kinesis stream with data stored in S3, Dynamo DB tables, and HDFS. You can write queries that join clickstream data from Kinesis with advertising campaign information stored in a DynamoDB table to identify the most effective categories of ads that are displayed on particular websites.
* Ad-hoc Queries: You can periodically load data from Kinesis into HDFS and make it available as a local Impala table for fast, interactive, analytic queries.

Q: What EMR AMI version do I need to be able to use the connector?

You need to use EMR’s AMI version 3.0.4 and later.

Q: Is this connector a stand-alone tool?

No, it is a built in component of the Amazon distribution of Hadoop and is present on EMR AMI versions 3.0.4 and later. Customer simply needs to spin up a cluster with AMI version 3.0.4 or later to start using this feature.

Q: What data format is required to allow EMR to read from a Kinesis stream?

The EMR Kinesis integration is not data format-specific. You can read data in any format. Individual Kinesis records are presented to Hadoop as standard records that can be read using any Hadoop MapReduce framework. Individual frameworks like Hive, Pig and Cascading have built in components that help with serialization and deserialization, making it easy for developers to query data from many formats without having to implement custom code. For example, in Hive users can read data from JSON files, XML files and SEQ files by specifying the appropriate [Hive SerDe](http://blog.cloudera.com/blog/2012/12/how-to-use-a-serde-in-apache-hive/) when they define a table. Pig has a similar component called [Loadfunc/Evalfunc](http://wiki.apache.org/pig/UDFManual) and Cascading has a similar component called a [Tap](http://docs.cascading.org/cascading/2.1/userguide/html/ch03s05.html). Hadoop users can leverage the extensive ecosystem of Hadoop adapters without having to write format-specific code. You can also implement custom deserialization formats to read domain specific data in any of these tools.

Q: How do I analyze a Kinesis stream using Hive in EMR?

Create a table that references a Kinesis stream. You can then analyze the table like any other table in Hive. Please see our [tutorials](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-kinesis.html) for page more details.

Q: Using Hive, how do I create queries that combine Kinesis stream data with other data source?

First create a table that references a Kinesis stream. Once a Hive table has been created, you can join it with tables mapping to other data sources such as Amazon S3, Amazon Dynamo DB, and HDFS. This effectively results in joining data from Kinesis stream to other data sources.

Q: Is this integration only available for Hive?

No, you can use Hive, Pig, MapReduce, Hadoop Streaming, and Cascading.

Q: How do I setup scheduled jobs to run on a Kinesis stream?

The EMR Kinesis input connector provides features that help you configure and manage scheduled periodic jobs in traditional scheduling engines such as Cron. For example, you can develop a Hive script that runs every N minutes. In the configuration parameters for a job, you can specify a Logical Name for the job. The Logical Name is a label that will inform the EMR Kinesis input connector that individual instances of the job are members of the same periodic schedule. The Logical Name allows the process to take advantage of iterations, which are explained next.

Since MapReduce is a batch processing framework, to analyze a Kinesis stream using EMR, the continuous stream is divided in to batches. Each batch is called an Iteration. Each Iteration is assigned a number, starting with 0. Each Iteration’s boundaries are defined by a start sequence number and end sequence number. Iterations are then processed sequentially by EMR.

In the event of an attempt’s failure, the EMR Kinesis input connector will re-try the iteration within the Logical Name from the known start sequence number of the iteration. This functionality ensures that successive attempts on the same iteration will have precisely the same input records from the Kinesis stream as the previous attempts. This guarantees idempotent (consistent) processing of a Kinesis stream.

You can specify Logical Names and Iterations as runtime parameters in your respective Hadoop tools. For example, in the [tutorial](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-kinesis.html)section “Running queries with checkpoints”, the code sample shows a scheduled Hive query that designates a Logical Name for the query and increments the iteration with each successive run of the job.

Additionally, a sample cron scheduling script is provided in the [tutorials](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-kinesis.html).

Q: Where is the metadata for Logical Names and Iterations stored?

The metadata that allows the EMR Kinesis input connector to work in scheduled periodic workflows is stored in Amazon DynamoDB. You must provision an Amazon Dynamo DB table and specify it as an input parameter to the Hadoop Job. It is important that you configure appropriate IOPS for the table to enable this integration. Please refer to the getting started [tutorial](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-kinesis.html) for more information on setting up your Amazon Dynamo DB table.

Q: What happens when an iteration processing fails?

Iterations identifiers are user-provided values that map to specific boundary (start and end sequence numbers) in a Kinesis stream. Data corresponding to these boundaries is loaded in the Map phase of the MapReduce job. This phase is managed by the framework and will be automatically re-run (three times by default) in case of job failure. If all the retries fail, you would still have options to retry the processing starting from last successful data boundary or past data boundaries. This behavior is controlled by providing kinesis.checkpoint.iteration.no parameter during processing. Please refer to the getting started [tutorial](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-kinesis.html) for more information on how this value is configured for different tools in the Hadoop ecosystem.

Q: Can I run multiple queries on the same iteration?

Yes, you can specify a previously run iteration by setting the kinesis.checkpoint.iteration.no parameter in successive processing. The implementation ensures that successive runs on the same iteration will have precisely the same input records from the Kinesis stream as the previous runs.

Q: What happens if records in an Iteration expire from the Kinesis stream?

In the event that the beginning sequence number and/or end sequence number of an iteration belong to records that have expired from the Kinesis steam, the Hadoop job will fail. You would need to use a different Logical Name to process data from the beginning of the Kinesis stream.

Q: Can I push data from EMR into Kinesis stream?

No. The EMR Kinesis connector currently does not support writing data back into a Kinesis stream.

Q: Does the EMR Hadoop input connector for Kinesis enable continuous stream processing?

The Hadoop MapReduce framework is a batch processing system. As such, it does not support continuous queries. However there is an emerging set of Hadoop ecosystem frameworks like Twitter Storm and Spark Streaming that enable to developers build applications for continuous stream processing. A Storm connector for Kinesis is available at on GitHub [here](https://github.com/awslabs/kinesis-storm-spout) and you can find a tutorial explaining how to setup Spark Streaming on EMR and run continuous queries [here](http://aws.amazon.com/articles/4926593393724923).

Additionally, developers can utilize the Kinesis client library to develop real-time stream processing applications. You can find more information on developing custom Kinesis applications in the Kinesis documentation [here](http://docs.aws.amazon.com/kinesis/latest/dev/step-three-build-an-app.html).

Q: Can I specify access credential to read a Kinesis stream that is managed in another AWS account?

Yes. You can read streams from another AWS account by specifying the appropriate access credentials of the account that owns the Kinesis stream. By default, the Kinesis connector utilizes the user-supplied access credentials that are specified when the cluster is created. You can override these credentials to access streams from other AWS Accounts by setting the kinesis.accessKey and kinesis.secretKey parameters. The following examples show how to set the kinesis.accessKey and kinesis.secretKey parameters in Hive and Pig.

Code sample for Hive:  
...  
STORED BY  
'com.amazon.emr.kinesis.hive.KinesisStorageHandler'  
TBLPROPERTIES(  
"kinesis.accessKey"="AwsAccessKey",  
"kinesis.secretKey"="AwsSecretKey",  
);

Code sample for Pig:  
…  
raw\_logs = LOAD 'AccessLogStream' USING com.amazon.emr.kinesis.pig.Kin  
esisStreamLoader('kinesis.accessKey=AwsAccessKey', 'kinesis.secretKey=AwsSecretKey'  
) AS (line:chararray);

Q: Can I run multiple parallel queries on a single Kinesis Stream? Is there a performance impact?

Yes, a customer can run multiple parallel queries on the same stream by using separate logical names for each query. However, reading from a shard within a Kinesis stream is subjected to a rate limit of of 2MB/sec. Thus, if there are N parallel queries running on the same stream, each one would get roughly (2/N) MB/sec egress rate per shard on the stream. This may slow down the processing and in some cases fail the queries as well.

Q: Can I join and analyze multiple Kinesis streams in EMR?

Yes, for example in Hive, you can create two tables mapping to two different Kinesis streams and create joins between the tables.

Q: Does the EMR Kinesis connector handle Kinesis scaling events, such as merge and split events?

Yes. The implementation handles split and merge events. The Kinesis connector ties individual Kinesis shards (the logical unit of scale within a Kinesis stream) to Hadoop MapReduce map tasks. Each unique shard that exists within a stream in the logical period of an Iteration will result in exactly one map task. In the event of a shard split or merge event, Kinesis will provision new unique shard Ids. As a result, the MapReduce framework will provision more map tasks to read from Kinesis. All of this is transparent to the user.

Q: What happens if there are periods of “silence” in my stream?

The implementation allows you to configure a parameter called kinesis.nodata.timeout. For example, consider a scenario where kinesis.nodata.timeout is set to 2 minutes and you want to run a Hive query every 10 minutes. Additionally, consider some data has been written to the stream since the last iteration (10 minutes ago). However, currently no new records are arriving, i.e. there is a silence in the stream. In this case, when the current iteration of the query launches, the Kinesis connector would find that no new records are arriving. The connector will keep polling the stream for 2 minutes and if no records arrive for that interval then it will stop and process only those records that were already read in the current batch of stream. However, if new records start arriving before kinesis.nodata.timeout interval is up, then the connector will wait for an additional interval corresponding to a parameter called kinesis.iteration.timeout. Please look at the [tutorials](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-kinesis.html) to see how to define these parameters.

Q: How do I debug a query that continues to fail in each iteration?

In the event of a processing failure, you can utilize the same tools they currently do when debugging Hadoop Jobs. Including the Amazon EMR web console, which helps identify and access error logs. More details on debugging an EMR job can be found [here](http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/emr-plan-debugging.html).

Q: What happens if I specify a DynamoDB table that I don’t have access to?

The job would fail and the exception would show up in error logs for the job.

Q: What happens if job doesn’t fail but checkpointing to DynamoDB fails?

The job would fail and the exception would show up in error logs for the job.

Q: How do I maximize the read throughput from Kinesis stream to EMR?

Throughput from Kinesis stream increases with instance size used and record size in the Kinesis stream. We recommend that you use m1.xlarge and above for both master and core nodes for this feature.