S3-simple storage service

* S3 is one of the first services that has been produced by aws.
* S3 stands for Simple Storage Service.
* S3 provides developers and IT teams with secure, durable, highly scalable object storage.
* It is easy to use with a simple web services interface to store and retrieve any amount of data from anywhere on the web.

What is S3?

* S3 is a safe place to store the files.
* It is Object-based storage, i.e., you can store the images, word files, pdf files, etc.
* The files which are stored in S3 can be from 0 Bytes to 5 TB.
* It has unlimited storage means that you can store the data as much you want.
* Files are stored in Bucket. A bucket is like a folder available in S3 that stores the files.
* S3 is a universal namespace, i.e., the names must be unique globally. Bucket contains a DNS address. Therefore, the bucket must contain a unique name to generate a unique DNS address.

If you create a bucket, URL look like:

AWS S3

Advantages of Amazon S3



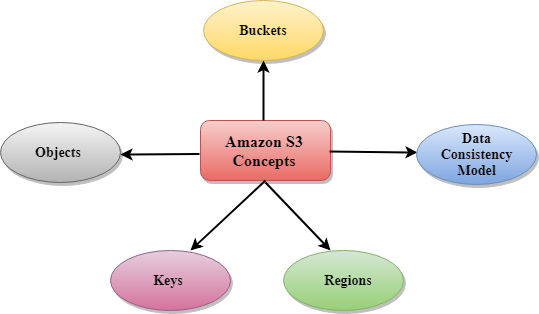
* **Create Buckets:** Firstly, we create a bucket and provide a name to the bucket. Buckets are the containers in S3 that stores the data. Buckets must have a unique name to generate a unique DNS address.
* **Storing data in buckets:** Bucket can be used to store an infinite amount of data. You can upload the files as much you want into an Amazon S3 bucket, i.e., there is no maximum limit to store the files. Each object can contain upto 5 TB of data. Each object can be stored and retrieved by using a unique developer assigned-key.
* **Download data:** You can also download your data from a bucket and can also give permission to others to download the same data. You can download the data at any time whenever you want.
* **Permissions:** You can also grant or deny access to others who want to download or upload the data from your Amazon S3 bucket. Authentication mechanism keeps the data secure from unauthorized access.
* **Standard interfaces:** S3 is used with the standard interfaces REST and SOAP interfaces which are designed in such a way that they can work with any development toolkit.
* **Security:** Amazon S3 offers security features by protecting unauthorized users from accessing your data.

S3 is a simple key-value store

**S3 is object-based. Objects consist of the following:**

* **Key:** It is simply the name of the object. For example, hello.txt, spreadsheet.xlsx, etc. You can use the key to retrieve the object.
* **Value:** It is simply the data which is made up of a sequence of bytes. It is actually a data inside the file.
* **Version ID:** Version ID uniquely identifies the object. It is a string generated by S3 when you add an object to the S3 bucket.
* **Metadata:** It is the data about data that you are storing. A set of a name-value pair with which you can store the information regarding an object. Metadata can be assigned to the objects in Amazon S3 bucket.
* **Subresources:** Subresource mechanism is used to store object-specific information.
* **Access control information:** You can put the permissions individually on your files.

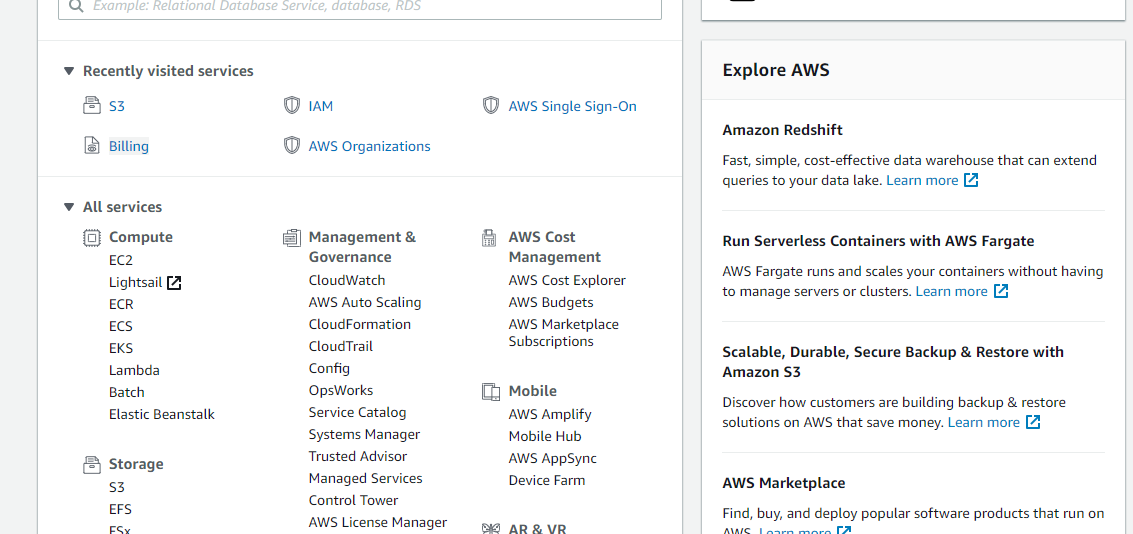
Amazon S3 Concepts



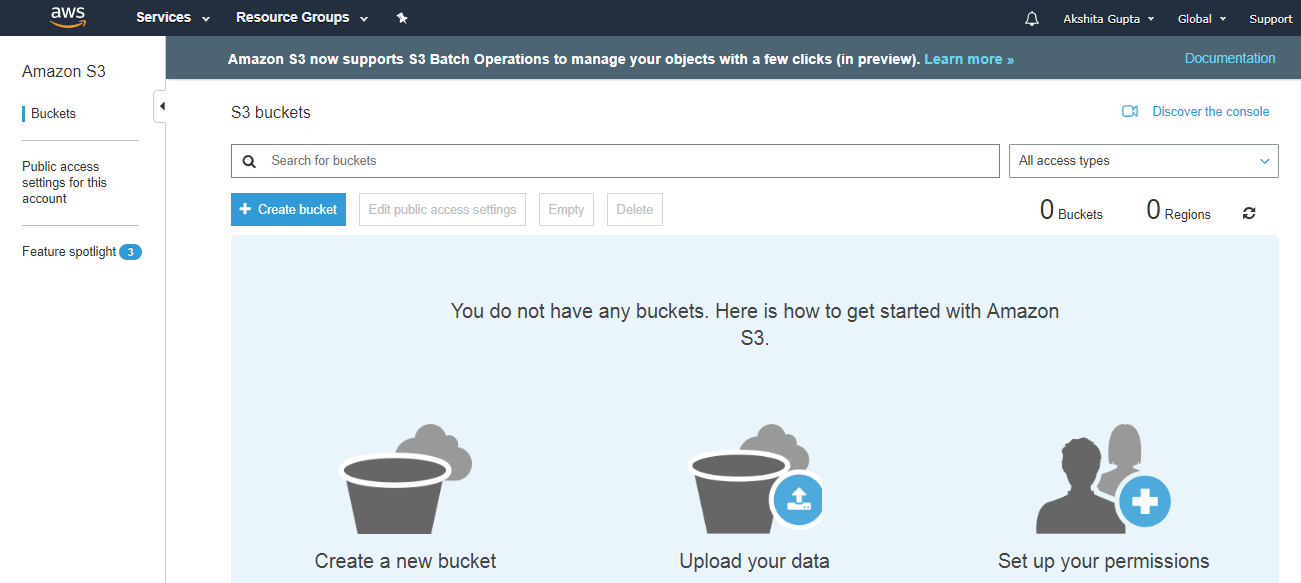
* Buckets
* Objects
* Keys
* Regions
* Data Consistency Model
* **Buckets**
  + A bucket is a container used for storing the objects.
  + Every object is incorporated in a bucket.
  + For example, if the object named photos/tree.jpg is stored in the treeimage bucket, then it can be addressed by using the URL http://treeimage.s3.amazonaws.com/photos/tree.jpg.
  + A bucket has no limit to the amount of objects that it can store. No bucket can exist inside of other buckets.
  + S3 performance remains the same regardless of how many buckets have been created.
  + The AWS user that creates a bucket owns it, and no other AWS user cannot own it. Therefore, we can say that the ownership of a bucket is not transferrable.
  + The AWS account that creates a bucket can delete a bucket, but no other AWS user can delete the bucket.
* **Objects**
  + Objects are the entities which are stored in an S3 bucket.
  + An object consists of object data and metadata where metadata is a set of name-value pair that describes the data.
  + An object consists of some default metadata such as date last modified, and standard HTTP metadata, such as Content type. Custom metadata can also be specified at the time of storing an object.
  + It is uniquely identified within a bucket by key and version ID.
* **Key**
  + A key is a unique identifier for an object.
  + Every object in a bucket is associated with one key.
  + An object can be uniquely identified by using a combination of bucket name, the key, and optionally version ID.
  + For example, in the URL http://jtp.s3.amazonaws.com/2019-01-31/Amazons3.wsdl where "jtp" is the bucket name, and key is "2019-01-31/Amazons3.wsdl"
* **Regions**
  + You can choose a geographical region in which you want to store the buckets that you have created.
  + A region is chosen in such a way that it optimizes the latency, minimize costs or address regulatory requirements.
  + Objects will not leave the region unless you explicitly transfer the objects to another region.

Creating an S3 Bucket

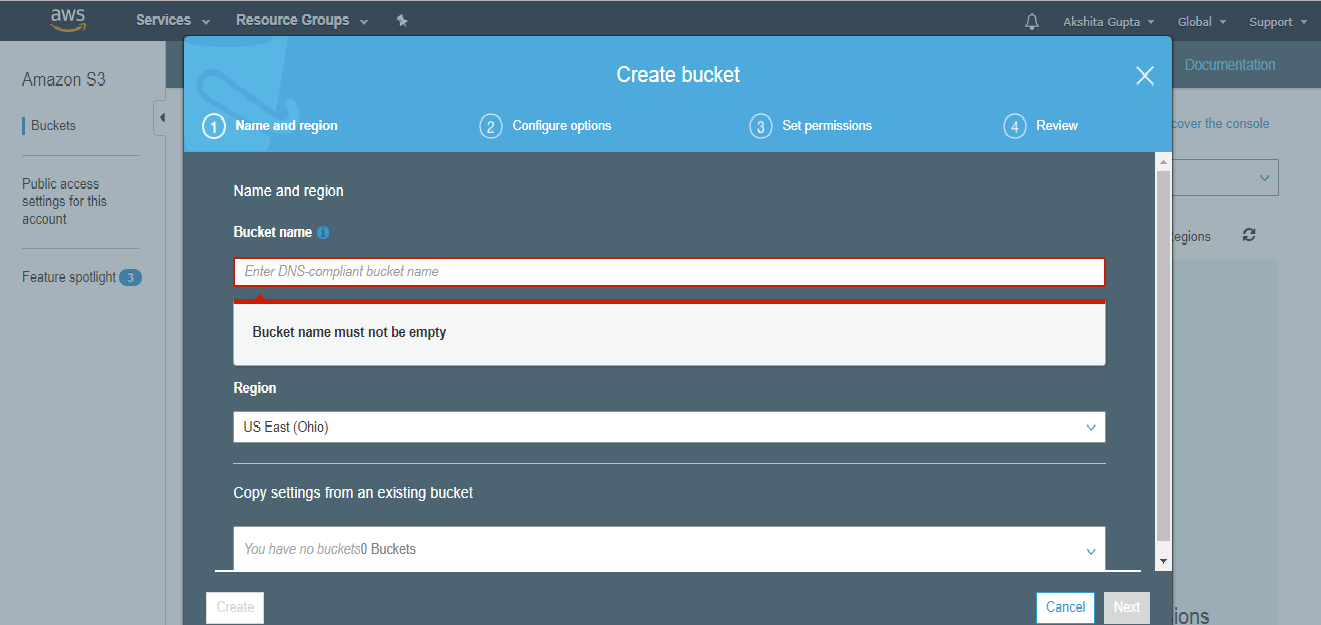
* Sign in to the AWS Management console. After sign in, the screen appears is shown below:



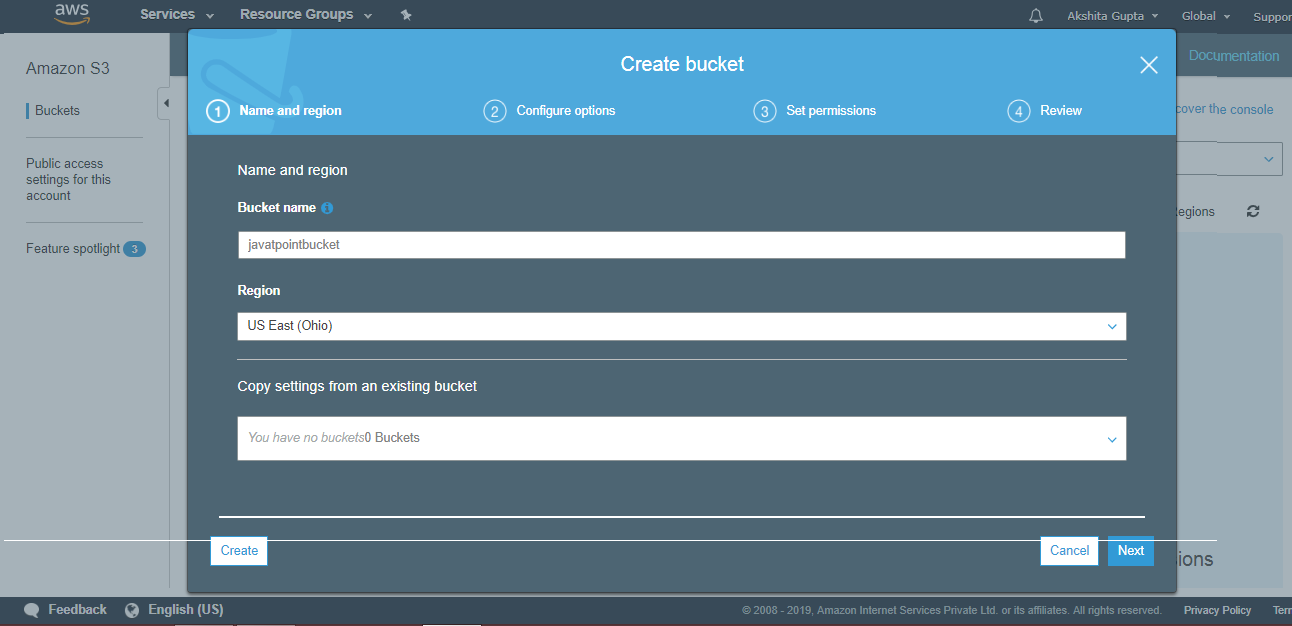
* Move to the S3 services. After clicking on S3, the screen appears is shown below:



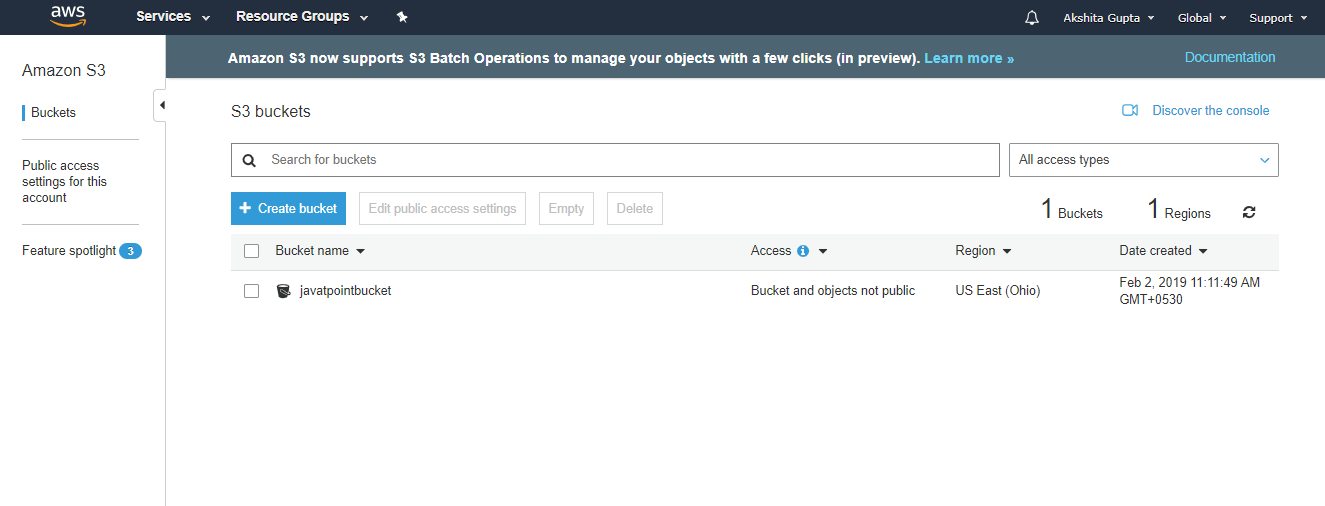
* To create an S3 bucket, click on the "Create bucket". On clicking the "Create bucket" button, the screen appears is shown below:



* Enter the bucket name which should look like DNS address, and it should be resolvable. A bucket is like a folder that stores the objects. A bucket name should be unique. A bucket name should start with the lowercase letter, must not contain any invalid characters. It should be 3 to 63 characters long.

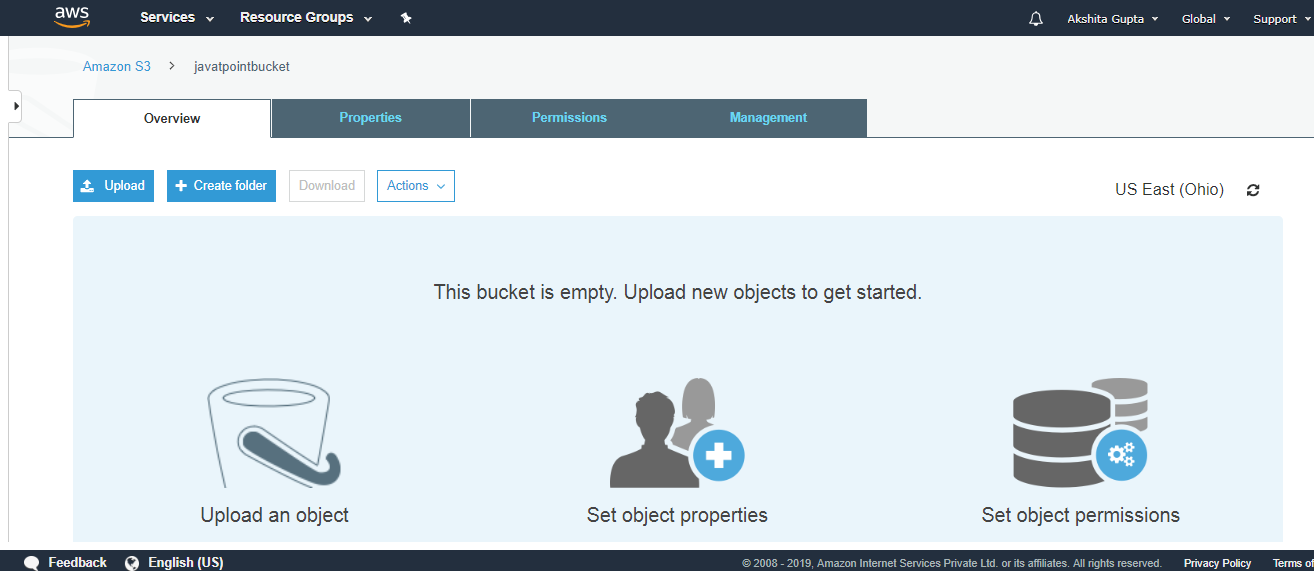


* Click on the "Create" button. Now, the bucket is created.

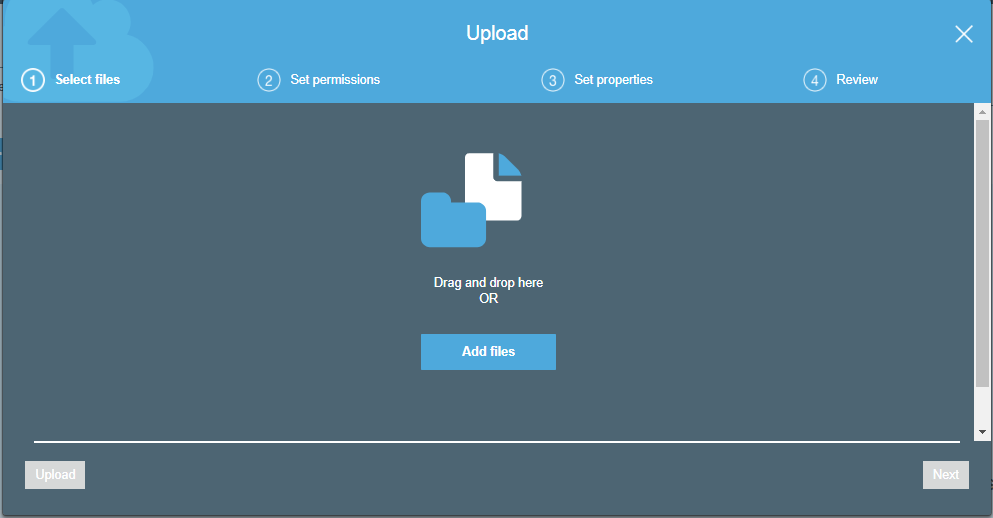


We have seen from the above screen that bucket and its objects are not public as by default, all the objects are private.

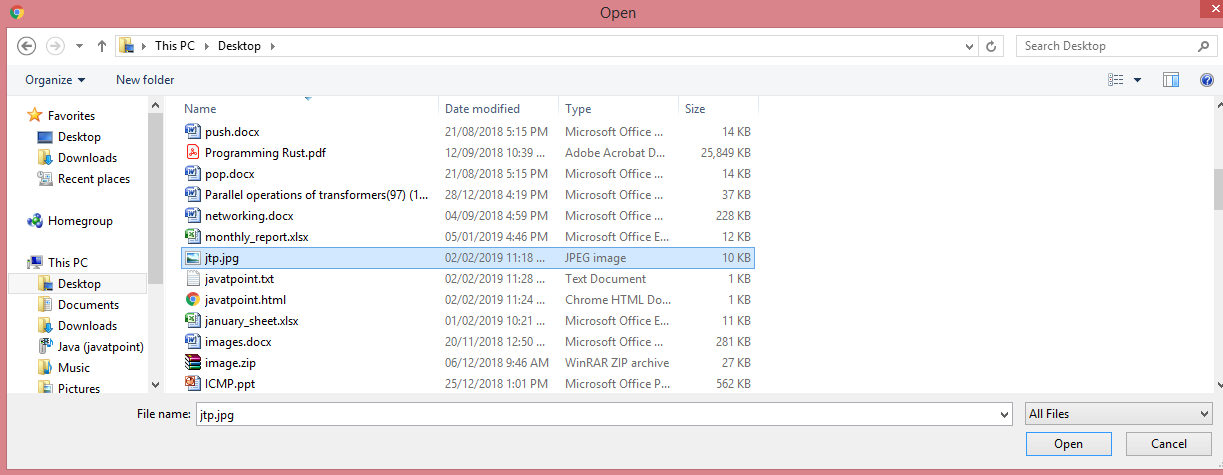
* Now, click on the "javatpointbucket" to upload a file in this bucket. On clicking, the screen appears is shown below:



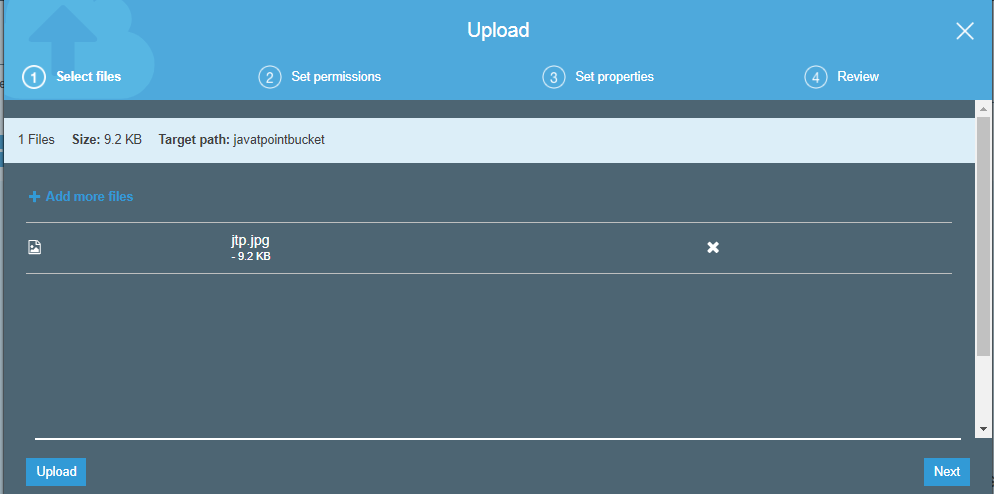
* Click on the "Upload" button to add the files to your bucket.



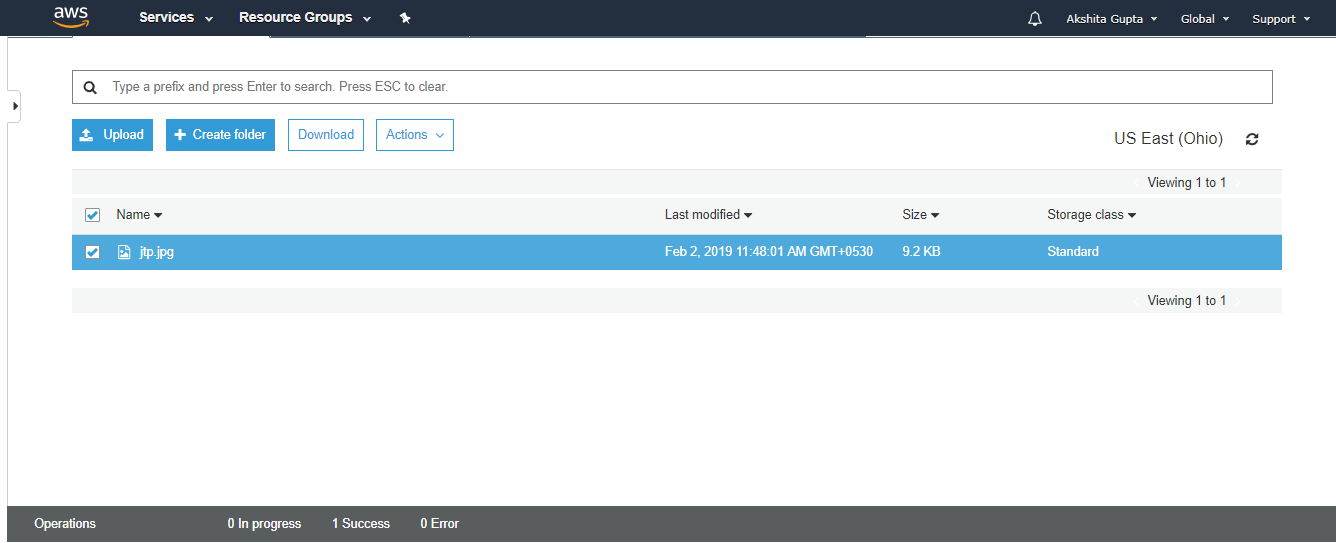
* Click on the "Add files" button.



* Add the **jtp.jpg** file.

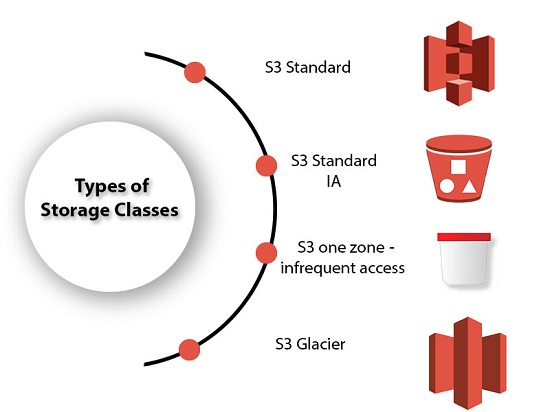


* Click on the "upload" button.



From the above screen, we observe that the "jtp.jpg" has been successfully uploaded to the bucket "javatpoint".

AWS Storage Classes



* S3 storage classes are used to assist the concurrent loss of data in one or two facilities.
* S3 storage classes maintain the integrity of the data using checksums.
* S3 provides lifecycle management for the automatic migration of objects for cost savings.

**S3 contains four types of storage classes:**

* S3 Standard
* S3 Standard IA
* S3 one zone-infrequent access
* S3 Glacier

S3 Standard

* Standard storage class stores the data redundantly across multiple devices in multiple facilities.
* It is designed to sustain the loss of 2 facilities concurrently.
* Standard is a default storage class if none of the storage class is specified during upload.
* It provides low latency and high throughput performance.
* It designed for 99.99% availability and 99.999999999% durability

S3 Standard IA

* IA stands for infrequently accessed.
* Standard IA storage class is used when data is accessed less frequently but requires rapid access when needed.
* It has a lower fee than S3, but you will be charged for a retrieval fee.
* It is designed to sustain the loss of 2 facilities concurrently.
* It is mainly used for larger objects greater than 128 KB kept for atleast 30 days.
* It provides low latency and high throughput performance.
* It designed for 99.99% availability and 99.999999999% durability

S3 one zone-infrequent access

* S3 one zone-infrequent access storage class is used when data is accessed less frequently but requires rapid access when needed.
* It stores the data in a single availability zone while other storage classes store the data in a minimum of three availability zones. Due to this reason, its cost is 20% less than Standard IA storage class.
* It is an optimal choice for the less frequently accessed data but does not require the availability of Standard or Standard IA storage class.
* It is a good choice for storing the backup data.
* It is cost-effective storage which is replicated from other AWS region using S3 Cross Region replication.
* It has the same durability, high performance, and low latency, with a low storage price and low retrieval fee.
* It designed for 99.5% availability and 99.999999999% durability of objects in a single availability zone.
* It provides lifecycle management for the automatic migration of objects to other S3 storage classes.
* The data can be lost at the time of the destruction of an availability zone as it stores the data in a single availability zone.

S3 Glacier

* S3 Glacier storage class is the cheapest storage class, but it can be used for archive only.
* You can store any amount of data at a lower cost than other storage classes.
* S3 Glacier provides three types of models:
  + **Expedited:** In this model, data is stored for a few minutes, and it has a very higher fee.
  + **Standard:** The retrieval time of the standard model is 3 to 5 hours.
  + **Bulk:** The retrieval time of the bulk model is 5 to 12 hours.
* You can upload the objects directly to the S3 Glacier.
* It is designed for 99.999999999% durability of objects across multiple availability zones.

**Performance across the Storage classes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **S3 Standard** | **S3 Standard IA** | **S3 One Zone-IA** | **S3 Glacier** |
| **Designed for durability** | 99.99999999% | 99.99999999% | 99.99999999% | 99.99999999% |
| **Designed for availability** | 99.99% | 99.9% | 99.5% | N/A |
| **Availability SLA** | 99.9% | 99% | 99% | N/A |
| **Availability zones** | >=3 | >=3 | 1 | >=3 |
| **Minimum capacity charge per object** | N/A | 128KB | 128KB | 40KB |
| **Minimum storage duration charge** | N/A | 30 days | 30 days | 90 days |
| **Retrieval fee** | N/A | per GB retrieved | per GB retrieved | per GB retrieved |
| **First byte latency** | milliseconds | milliseconds | Milliseconds | Select minutes or hours |
| **Storage type** | Object | Object | Object | Object |
| **Lifecycle transitions** | Yes | Yes | Yes | Yes |

Important points to remember

* Buckets are a universal namespace, i.e., the bucket names must be unique.
* If uploading of an object to S3 bucket is successful, we receive a HTTP 200 code.
* S3, S3-IA, S3 Reduced Redundancy Storage are the storage classes.
* Encryption is of two types, i.e., Client Side Encryption and Server Side Encryption
* Access to the buckets can be controlled by using either ACL (Access Control List) or bucket policies.
* By default buckets are private and all the objects stored in a bucket are also private.

What is Load Balancer?

In the context of parallel computing, Load Balancing is the distribution of a set of tasks over different computing units (or related resources), to make the overall process easier to execute and much more efficient. Ensuring no single server bears too much of demand and evenly spreading the load, it improves the responsiveness and availability of applications or websites for the user.

Modern applications and websites cannot function without balancing the load on them. This is for the reason that such applications and sites serve millions of simultaneous requests from the end-users and have returned the correct text and images or the related data asked for, responsively, and reliably. Adding more servers was considered to be good practice for meeting such high volumes of traffic, so far.

But, the concept of balancing the load with a dedicated Load Balancer unit is a much more economical and effective way of ensuring the peak performance of the website or application and offering the end-user a great experience.

**History of Load Balancing**

The load balancing concept was initiated in 1990 with special hardware that was deployed to distribute traffic across a network. With the development of [Application Delivery Controllers](https://www.appviewx.com/products/adc/) (ADCs), load balancing became a better-secured convenience, offering uninterrupted access to applications even at peak times.

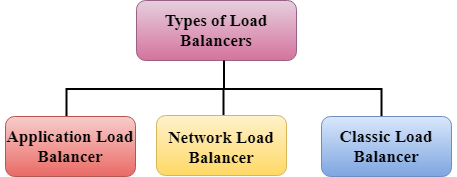
ADCs are categorized as Hardware Appliance, Virtual Appliance, and Software Native Load Balancers. In this era of cloud computing, the software-based ADCs are used to perform tasks as the hardware counterpart performs but with better scalability, functionality, and flexibility.

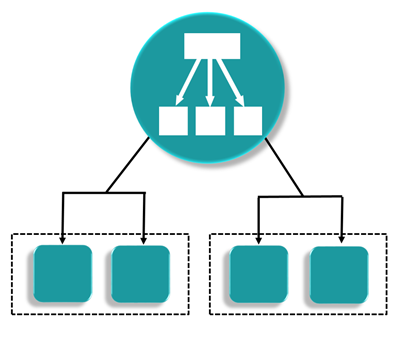
### https://www.appviewx.com/wp-content/uploads/2021/01/Understanding-the-functions-of-load-balancers.pngUnderstanding the functions of load balancers

In general, a Load Balancer acts as a ‘traffic controller’ for your server and directs the requests to an available one, capable of fulfilling the request efficiently. This ensures that requests are responded to fast and no server is over-stressed to degrade the performance.

In an organization’s attempt to meet the application demands, Load Balancer assists in deciding which server can efficiently handle the requests. This creates a better user experience.

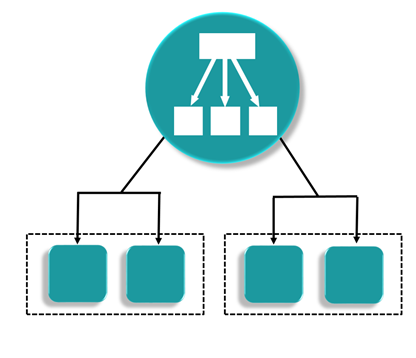
By helping servers move data efficiently, information flow between the server and endpoint device is also managed by the load balancer. It also assesses the request-handling health of the server, and if necessary, Load Balancer removes the unhealthy server until it is restored.  
As the servers can also be physical or virtual, a load balancer can also be a hardware appliance or a software-based virtual one. When a server goes down, the requests are directed to the remaining servers and when a new server gets added, the requests automatically start getting transferred to it.



Application Load Balancer

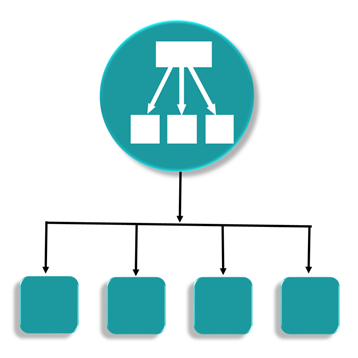
* An Amazon Web Services (AWS) launched a new load balancer known as an Application load balancer (ALB) on August 11, 2016.
* It is used to direct user traffic to the public AWS cloud.
* It identifies the incoming traffic and forwards it to the right resources. For example, if a URL has /**API** extensions, then it is routed to the appropriate application resources.
* It is operated at Layer 7 of the OSI Model.
* It is best suited for load balancing of HTTP and HTTPs traffic.
* Application load balancers are intelligent, sending specific requests to specific web servers.
* If we take an example of TESLA. We have three models of TESLA, i.e., TESLA Model X, TESLA Model S, and TESLA Model 3 and TESLAs have onboard computing facility. You will have a group of web servers that serve the Model X, a group of web servers that serve the Model S, and similarly for Model 3. We have one Load balance that checks whether the incoming traffic comes from either Model X, Model S or Model 3, and then sends it to the intended group of servers.

Network Load Balancer



* It is operated at the Layer 4 of the OSI model.
* It makes routing decisions at the transport layer (TCP/SSL), and it can handle millions of requests per second.
* When a load balancer receives a connection, it then selects a target from the target group by using a flow hash routing algorithm. It opens the TCP connection to the selected target of the port and forwards the request without modifying the headers.
* It is best suited for load balancing the TCP traffic when high performance is required.

Classic Load Balancer



* It is operated at Layer 4 of the OSI model.
* It routes the traffic between clients and backend servers based on IP address.
* For example, an Elastic Load balancer receives a request from a client on TCP port 80, it will then routes the request to a specified port of backend servers. The port on which the Load Balancer routes to the target server will be having port number 80. The backend server will then send the requested data back to the ELB, which will then forward the Backend server reply to the client. According to the client's perspective, the request has been fulfilled by the ELB, not by the backend server.
* Classic Load balancers are legacy Elastic load balancers.
* It can also be used for load balancing the HTTP or HTTPs traffic and use layer 7-specific features, such as X-forwarded and sticky sessions.
* You can also use the Layer 4 load balancing for applications that rely purely on the TCP protocol.

Load Balancer Errors

**Classic Load Balancer**

If you get an error 504, this is a gateway timeout error. A Load balancer is still available, but it has a problem in communicating with the EC2 instance. If your application stops responding, the ELB (Classic Load Balancer) responds with a 504 error. This means that the application is having issues and it could be either at the web server layer or the Database layer.

In order to troubleshoot where the application is failing, and scale it up or out where possible.