Amazon Web Services and Feb 28 outage

Overview presented by Divya

Amazon S3

- Amazon S3 : store and retrieve any amount of data, at any time, from anywhere on web.
- Amazon S3 service:
  - **Create Buckets** – Create and name a bucket that stores data. Buckets are fundamental container in Amazon S3 for data storage.
  - **Store data in Buckets** – Store an infinite amount of data in a bucket. Each object can contain up to **5 TB of data**. Stored and retrieved using a unique developer-assigned key.
  - **Download data** – any time you like or allow others to do same.
  - **Permissions** – Grant or deny access to others who want to upload or download data into your Amazon S3 bucket. Grant upload and download permissions to three types of users. Authentication mechanisms can help keep data secure from unauthorized access.
  - **Standard interfaces** – Use standards-based REST and SOAP interfaces designed to work with any Internet-development toolkit.

S3 functions

- Create a Bucket – Create and name your bucket.
- Write an Object – Store data by creating or overwriting an object. When you write an object, you specify a unique key in namespace of your bucket. This is also a good time to specify any access control you want on object.
- Read an Object – Read data back. You can download data via HTTP or Bit Torrent.
- Deleting an Object – Delete some of your data.
- **Listing Keys** – List keys contained in one of your buckets. You can filter key list based on a prefix.

Components of S3

Buckets: Container for objects stored in Amazon S3. Every object is contained in a bucket. For example, ifobject named photos/puppy.jpg is stored injohnsmith bucket, then it is addressable using URL http://johnsmith.s3.amazonaws.com/photos/puppy.jpg

- Organize Amazon S3 namespace at highest level
- Configure buckets for specific region.

Objects: Fundamental entities stored in Amazon S3. Consist of object data and metadata.

- Data portion is opaque to Amazon S3.
- **Metadata is a set of name-value pairs** that describe object.
- Meta data: date last modified, and standard HTTP metadata, such as Content-Type. An object is uniquely identified within a bucket by a key (name) and a version ID.

Keys: **unique identifier** for an object within a bucket.

- Amazon S3 is basic data map between **"bucket + key + version"** and object itself.
- Every object in Amazon S3 is uniquely addressed through combination of web service endpoint, bucket name, key, and optionally, a version. For example, in URL http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wsdl, "doc" is name of bucket and "2006-03-01/AmazonS3.wsdl" iskey.

Amazon S3 Data Consistency Model

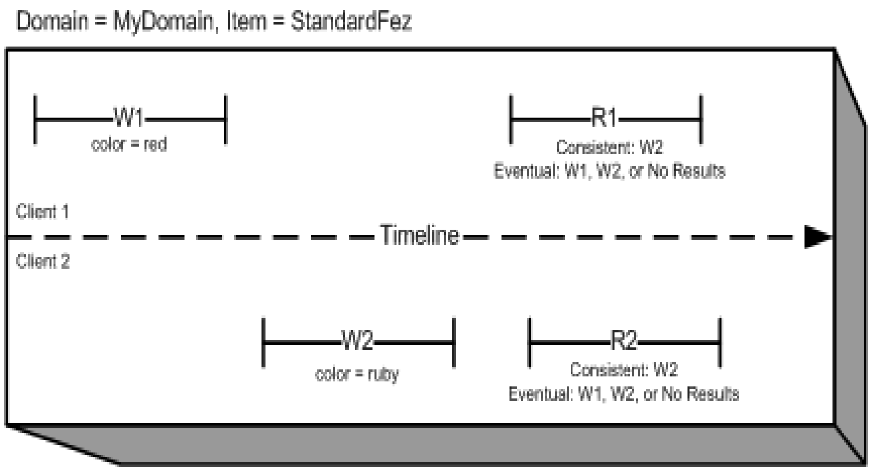
- Read-after-write consistency for PUTS of new objects in your S3 bucket in all regions with one caveat (HEAD or GET request to key name (to find if object exists) before creating object)
- Amazon S3 provides **eventual consistency for read-after-write.**
- Amazon S3 offers eventual consistency for overwrite PUTS and DELETES in all regions.
- **Updates to a single key are atomic.** For example, if you PUT to an existing key, a subsequent read might return old data or updated data, but it will never write corrupted or partial data.

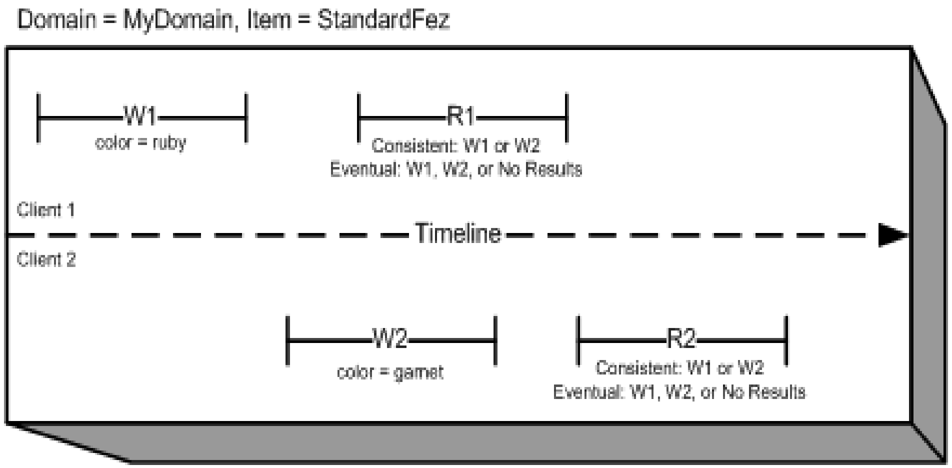
Consistency Model

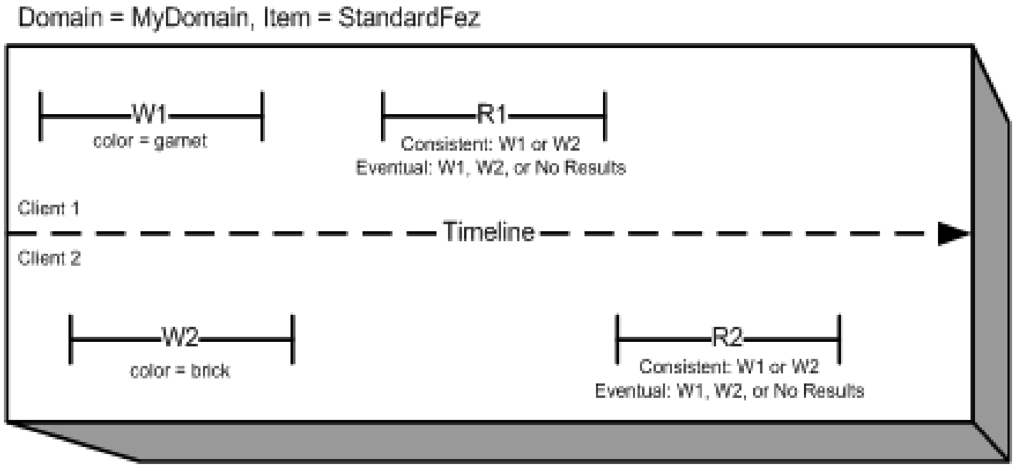
- **Amazon S3 achieves high availability by replicating data across multiple servers within Amazon's data centers**. If a PUT request is successful, data is safely stored.
- However, information about changes must replicate across Amazon S3, which can take some time, leading to:
  - A process writes a new object to Amazon S3 and immediately lists keys within its bucket. Until change is fully propagated, object might not appear in list.
  - A process replaces an existing object and immediately attempts to read it. Until change is fully propagated, Amazon S3 might return prior data.
  - A process deletes an existing object and immediately attempts to read it. Until deletion is fully propagated, Amazon S3 might return deleted data.
  - A process deletes an existing object and immediately lists keys within its bucket. Until deletion is fully propagated, Amazon S3 might list deleted object.

Bucket policies

- Only bucket owner is allowed to associate a policy with a bucket. Policies, written in access policy language, allow or deny requests based on:
- Amazon S3 bucket operations (such as PUT ?acl), and object operations (such as PUT Object, or GET Object)
- An account can control access based on specific Amazon S3 operations, such as GetObject, GetObjectVersion, DeleteObject, or DeleteBucket.
- conditions can be such things as IP addresses, IP address ranges in CIDR notation, dates, user agents, HTTP referrer and transports (HTTP and HTTPS).


Domain = MyDomain, Item = StandardFez
W1
color = red
R1
Consistent: W2
Eventual: W1, W2, or No Results
Client 1
Timeline
Client 2
W2
color = ruby
R2
Consistent: W2
Eventual: W1, W2, or No Results


Domain = MyDomain, Item = StandardFez
W1
color = ruby
R1
Consistent: W1 or W2
Eventual: W1, W2, or No Results
Client 1
Timeline
Client 2
W2
color = garnet
R2
Consistent: W1 or W2
Eventual: W1, W2, or No Results

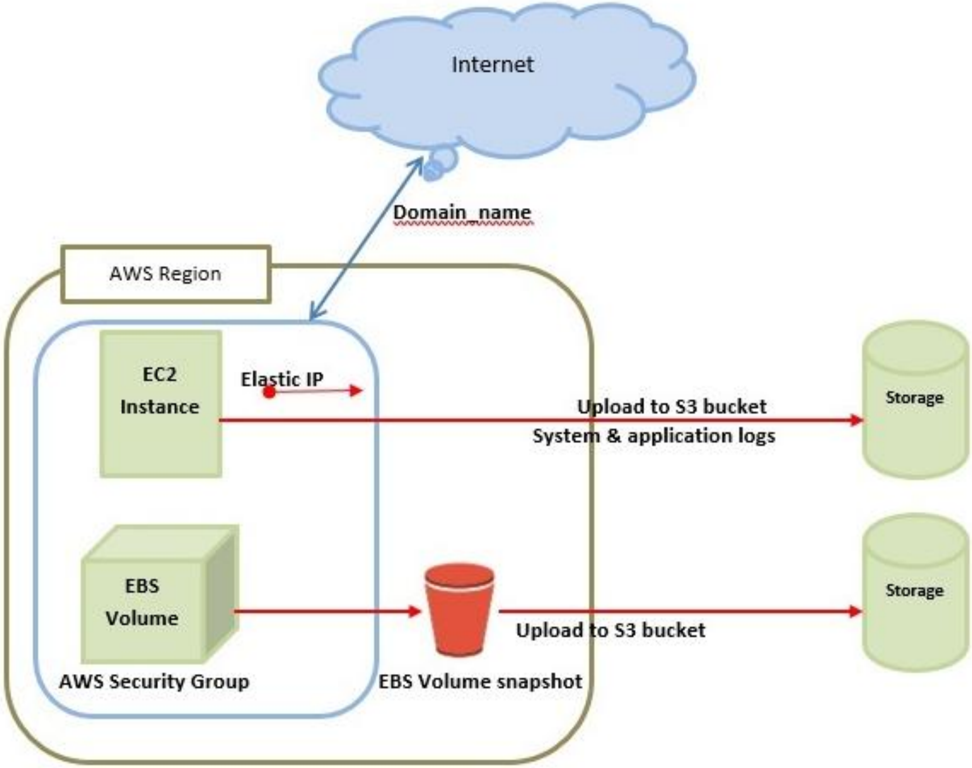

Domain = MyDomain, Item = StandardFez
W1
color = garnet
R1
Consistent: W1 or W2
Eventual: W1, W2, or No Results
Client 1
Timeline
Client 2
W2
color = brick
R2
Consistent: W1 or W2
Eventual: W1, W2, or No Results

Regions

Choose geographical region where Amazon S3 will store buckets you create. You might choose a region to optimize latency, minimize costs, or address regulatory requirements. Amazon S3 currently supports following regions:

- **US East (N. Virginia) Region** Uses Amazon S3 servers in Northern Virginia
- **US East (Ohio) Region** Uses Amazon S3 servers in Columbus Ohio
- **US West (N. California) Region** Uses Amazon S3 servers in Northern California
- **US West (Oregon) Region** Uses Amazon S3 servers in Oregon
- **Canada (Central) Region** Uses Amazon S3 servers in Canada
- **Asia Pacific (Mumbai) Region** Uses Amazon S3 servers in Mumbai
- **Asia Pacific (Seoul) Region** Uses Amazon S3 servers in Seoul
- **Asia Pacific (Singapore) Region** Uses Amazon S3 servers in Singapore
- **Asia Pacific (Sydney) Region** Uses Amazon S3 servers in Sydney
- **Asia Pacific (Tokyo) Region** Uses Amazon S3 servers in Tokyo
- **EU (Frankfurt) Region** Uses Amazon S3 servers in Frankfurt
- **EU (Ireland) Region** Uses Amazon S3 servers in Ireland
- **EU (London) Region** Uses Amazon S3 servers in London
- **South America (São Paulo) Region** Uses Amazon S3 servers in Sao Paulo

EC2



- **EC2** stands for Elastic Compute Cloud.
- EC2 allow users to use virtual machines of different configurations as per their requirement.
- Various configuration options, mapping of individual server, various pricing options, etc.
- AWS provides Elastic Load Balancing service, it distributes traffic to EC2 instances across multiple available sources, and dynamic addition and removal of Amazon EC2 hosts from load-balancing rotation.
- **Elastic Load Balancing** can dynamically grow and shrink load-balancing capacity to adjust to traffic demands and also support sticky sessions to address more advanced routing needs.

EC2 components

Amazon Cloud-front

- Content delivery, i.e. used to deliver website.
- Dynamic, static, and streaming content using a global network of edge locations.
- Requests for content at user's end are automatically routed to nearest edge location, which improves performance.

Security Management

- EC2 provides security groups, which is similar to an inbound network firewall, to specify protocols, ports, and source IP ranges that are allowed to reach EC2 instances.

Auto Scaling

- Difference between AWS cloud architecture and traditional hosting model is that AWS can dynamically scale web application fleet on demand to handle changes in traffic.
- In traditional hosting model, traffic forecasting models are generally used to provision hosts ahead of projected traffic

EC2 Components

Storage & Backups

- AWS cloud provides various options for storing, accessing, and backing up web application data and assets.
- **Amazon S3 (Simple Storage Service)** provides a simple web-services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on web.
- **Amazon EBS** is effective for data that needs to be accessed as block storage and requires persistence beyond life of running instance, such as database partitions and application logs.
- EBS volumes can be maximized up to 1 TB, and these volumes can be striped for larger volumes and increased performance. Provisioned IOPS volumes are designed to meet needs of database workloads that are sensitive to storage performance and consistency.
- **Amazon EBS currently supports up to 1,000 IOPS per volume**. We can stripe multiple volumes together to deliver thousands of IOPS per instance to an application.

Amazon RDS

Hosting RDMS on EC2 Instances

- Amazon RDS allows users to install RDBMS (Relational Database Management System) of your choice like MySQL, Oracle, SQL Server, DB2, etc. on an EC2 instance and can manage as required.
- Amazon EC2 uses Amazon EBS (Elastic Block Storage) similar to network-attached storage. All data and logs running on EC2 instances should be placed on Amazon EBS volumes, which will be available even if database host fails.
- **Amazon EBS** volumes automatically **provide redundancy within availability zone,** which increases availability of simple disks. Further if volume is not sufficient for databases needs, volume can be added to increase performance for our database.
- Using Amazon RDS, service provider manages storage and not on managing data.

AWS S3 outage

- **Lots of websites on Tuesday loaded slowly, or not at all.**
- reason was that Amazon Web Services, a server product from Amazon that powers a surprising number of sites and apps, experienced a major disruption in its S3 product that lasted several hours.
- Amazon gave a detailed postmortem on Thursday, blaming an employee who entered an input command "incorrectly," causing a larger set of servers than was expected to be removed from service.
- Those servers supported other AWS products and services, and it caused a chain reaction, which meant that certain critical systems had to be rebooted — and while they were restarting, Amazon's S3 wasn't working as normal.
- Sites including **Slack, Quora**, and even **US Securities and Exchange Commission were down** for much of Tuesday afternoon.
- service affected, Amazon S3, is used by developers to store files, so some websites may have stayed up but were slow to load or couldn't load any images.
- At one point **"dashboard,"** where Amazon tells its users which of its services are operational, wasn't working because ofS3 issue.

Amazon explanation for outage

- Northern Virginia (US-EAST-1) morning of February 28th.
- 9:37AM PST, **S3 team member executed a command** to remove a small number of servers for S3 **subsystems billing** process.
- Command was entered incorrectly and a larger set of servers was removed than intended.
- **Removed servers supported two other S3 subsystems.**
- 1. **index subsystem**, manages metadata and location information of all S3 objects in region, necessary to serve all **GET, LIST, PUT, and DELETE requests.**
- 2. **placement subsystem**, manages allocation of **new storage and requires index subsystem to be functioning properly to correctly operate placement** subsystem is used during PUT requests to allocate storage for new objects.
- Removing a significant portion of capacity a full restart.
- While these subsystems were being restarted, S3 was unable to service requests.
- Other AWS services in US-EAST-1 Region that rely on S3 for storage, including **S3 console, Amazon Elastic Compute Cloud (EC2) new instance launches, Amazon Elastic Block Store (EBS) volumes (when data was needed from a S3 snapshot), and AWS Lambda** were also impacted while S3 APIs were unavailable.

Recovery process

- Index subsystem was first of two affected subsystems that needed to be restarted.
- By 12:26PM PST, index subsystem had activated enough capacity to begin servicing S3 GET, LIST, and DELETE requests.
- By 1:18PM PST, index subsystem was fully recovered and GET, LIST, and DELETE APIs were functioning normally.
- S3 PUT API also required placement subsystem.
- placement subsystem began recovery when index subsystem was functional and finished recovery at 1:54PM PST. At this point, S3 was operating normally.
- Other AWS services that were impacted by this event began recovering.
- Some of these services had accumulated a backlog of work during S3 disruption and required additional time to fully recover.

Changes to S3

- Modified tool to remove capacity more slowly and added safeguards to prevent capacity from being removed when it will take any subsystem below its minimum required capacity level.
- Prevent an incorrect input from triggering a similar event in future.
- Auditing our other operational tools to ensure similar safety checks.
- Improve recovery time of key S3 subsystems using multiple
  - Involves **breaking services into small partitions called cells**. Teams can assess and thoroughly test recovery processes of even largest service or subsystem. As S3 has scaled, team has done considerable work to refactor parts of service into smaller cells to reduce blast radius and improve recovery.
  - During this event, recovery time of index subsystem still took longer than we expected. S3 team had planned further partitioning of index subsystem later this year.