I DIDN'T KNOW S3 COULD DO THAT!

Brian Klaas
brian.klaas@gmail.com
@brian_klaas

Why this talk?

S3 POWERS THE WEB

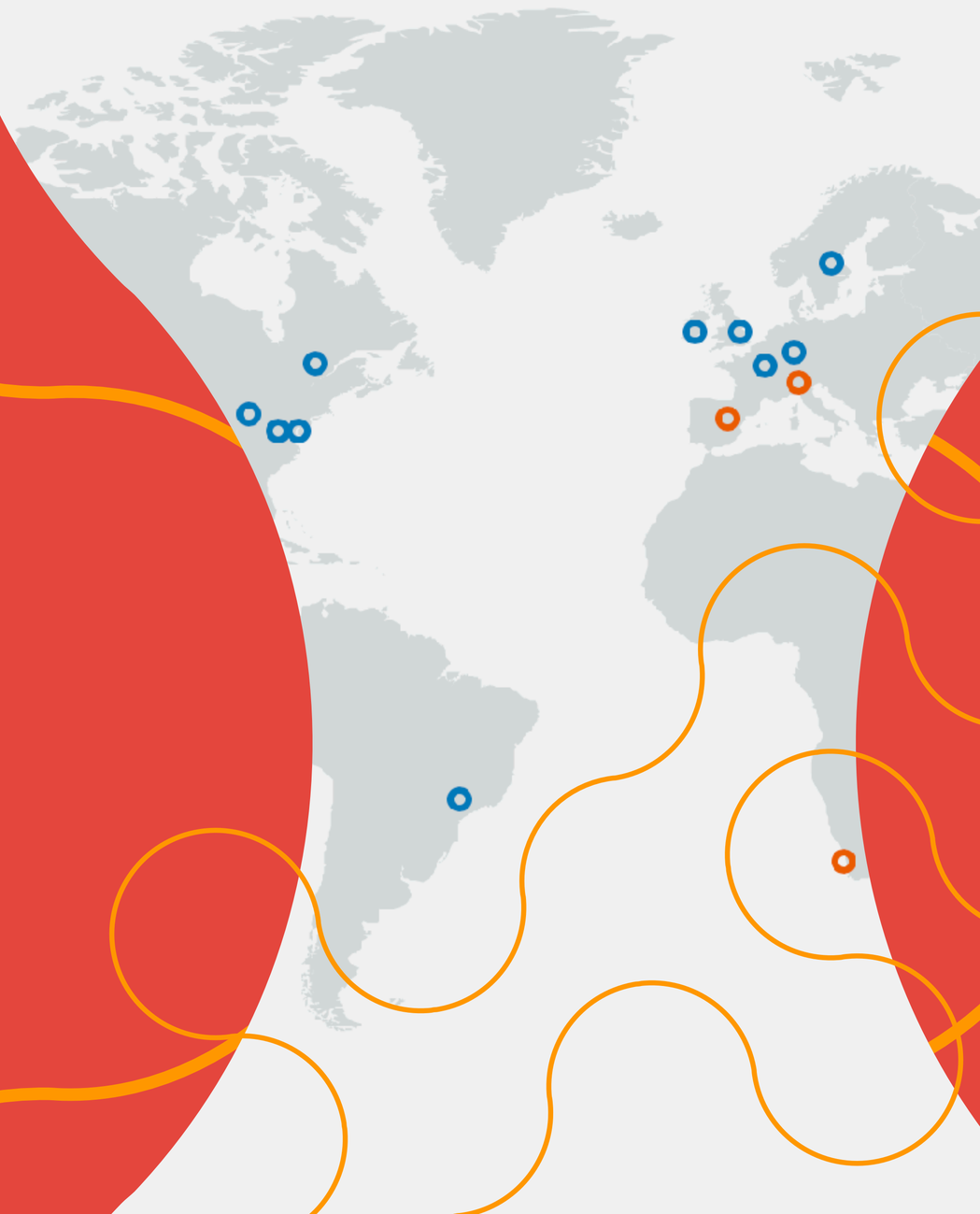








60 TERRABITS PER SECOND



CF/FILE INTEGRATION

- <cffile action="read" file="s3://accessKey:secretKey@somebucket/somefile.txt" variable="fileData" />
- fileWrite("s3://somebucket/somefile.txt", textOfFile);
- imageAsBinary = fileReadBinary("s3://userfilestorage/userImage.jpg");

Why this talk?

THERE IS SO MUCH MORE

download(**PresignedUrlDownloadRequest** presignedUrlDownloadRequest, **File** destinationFile)
Gets the object stored in Amazon S3 using a presigned url.

ableRequesterPays(**String** bucketName)
ws Amazon S3 bucket owner to enable the Requester Pays for the given bucket name.

ratePresignedUrl(**GeneratePresignedUrlRequest** req)
Re signed URL for accessing an Amazon S3 resource.

gen ignedUrl(**String** bucketName, **String** key, **Date** expiration)
Re gned URL for accessing an Amazon S3 resource.

ratePresignedUrl(**String** bucketName, **String** key, **Date** expiration, **HttpMethod** method)
rns a pre-signed URL for accessing an Amazon S3 resource.

getBucketAccelerateConfiguration(**GetBucketAccelerateConfigurationRequest** getBucketAccelerateConfiguration
Retrieves the accelerate configuration for the given bucket.

getBucketAccelerateConfiguration(**String** bucketName)
Retrieves the accelerate configuration for the given bucket.

getBucketAcl(**GetBucketAclRequest** getBucketAclRequest)
Gets the **AccessControlList** (ACL) for the specified Amazon S3 bucket.

getBucketAcl(**String** bucketName)
Gets the **AccessControlList** (ACL) for the specified Amazon S3 bucket.

getBucketAnalyticsConfiguration(**GetBucketAnalyticsConfigurationRequest** getBucketAnalyticsConfigurationRe
Gets an analytics configuration for the bucket (specified by the analytics configuration ID).

getBucketAnalyticsConfiguration(**String** bucketName, **String** id)
Gets an analytics configuration for the bucket (specified by the analytics configuration ID).

getBucketCrossOriginConfiguration(**GetBucketCrossOriginConfigurationRequest** getBucketCrossOriginConfigura
Gets the cross origin configuration for the specified bucket, or null if no configuration has been established.

getBucketCrossOriginConfiguration(**String** bucketName)
Gets the cross origin configuration for the specified bucket, or null if the specified bucket does not exist, or an empty list if no configuration has

getBucketEncryption(**GetBucketEncryptionRequest** getBucketEncryptionRequest)
Returns the server-side encryption configuration of a bucket.

getBucketEncryption(**String** bucketName)
Returns the server-side encryption configuration of a bucket.

getBucketInventoryConfiguration(**GetBucketInventoryConfigurationRequest** getBucketInventoryConfigurationRe
Returns an inventory configuration (identified by the inventory ID) from the bucket.

AWS JAVA SDK

Add to:

cfusion/lib

lucee-server/context/lib/

Why this talk?

AWS PLAYBOX APP

AWS Service Playbox

oDB

dentity Access Management (IAM)

Lambda Function Invocation

Rekognition

Simple Notification Service (SNS)

Simple Storage Service (S3)

Step Functions

Transcribe

Translate

github.com/brianklaas/awsPlaybox

COLDFUSION 2021

Native support for S3 and Azure Blob Services

No need to add the AWS Java SDK

LET’S TALK ABOUT

SECURITY

MONEY

POWER

...AND BEYOND

CONFIGURING THE AWS SERVICE IN COLDFUSION 2021

INSTALL THE AWSS3 PACKAGE

- CF Administrator
- cfpm.bat in CFHOME/cfusion/bin
- CommandBox: cfpm once the server is started

THREE WAYS TO CONFIGURE THE AWS SERVICE

- CF Administrator
- application.cfc
- Inline

CONFIGURE IN APPLICATION.CFC

```
void function onApplicationStart(){

    application.awsCredentials  = {

        "vendorName" : "AWS",

        "region" : "us-east-1",

        "accessKeyId" : "my access key",

       "secretAccessKey" : "my secret key"

    };

    application.s3Configuration  = {

        "serviceName" : "S3"

    };

    application.s3ServiceObject =
        getCloudService(application.awsCredentials,
        application.s3Configurtation);

}
```

BASIC PATTERN TO WORKING WITH S3 IN CF2021

- Get a reference to the bucket
- Create a struct of options
- Call a method of the bucket object using the options struct

EXAMPLE: UPLOADING A FILE

```
myBucket = application.s3ServiceObject.bucket(“your bucket name”);

uploadRequest  = {

    “srcFile”: “path/to/local/file.txt”,

    “key”: “fileName.txt”

};

uploadResponse = myBucket.uploadFile(uploadRequest);
```

SECURITY

Security

IDENTITY ACCESS MANAGEMENT (IAM)

ADOBE COLDFUSION 2018

```
"Effect": "Allow",
"Action": [
    "s3:GetObjectVersionTagging",
    "s3:CreateBucket",
    "s3:GetObjectAcl",
    "s3:GetBucketObjectLockConfiguration",
    "s3:GetObjectVersionAcl",
    "s3:PutBucketAcl",
    "s3:HeadBucket",
    "s3:DeleteObject",
    "s3:GetBucketPolicyStatus",
    "s3:GetObjectRetention",
    "s3:GetBucketWebsite",
    "s3:ListJobs",
    "s3:GetObjectLegalHold",
    "s3:GetBucketNotification",
    "s3:PutBucketCORS",
    "s3:DeleteBucketPolicy",
    "s3:GetReplicationConfiguration",
    "s3:ListMultipartUploadParts",
    "s3:PutObject",
```

YOUR OWN POLICY

```
"Effect": "Allow",
"Action": [
    "s3:PutObject",
    "s3:GetObjectAcl",
    "s3:GetObject",
    "s3:ListBucket",
    "s3:GetBucketAcl",
    "s3:DeleteObject"
],
```

MY IAM TALK FROM CFSUMMIT 2019

youtube.com/watch?v=ucn90XLDlPw

TIME-EXPIRING URLS

Security

TIME-EXPIRING URLS

NO MORE PUBLIC READ BUCKETS!

Security

TIME-EXPIRING URLS

NO MORE PUBLIC READ BUCKETS!

Security

TIME-EXPIRING URLS

- Secure your content
- Limit access to downloads
- Change file names on the fly
- Specify inline or attachment disposition
- Presigned PUT: Upload files directly to S3, bypassing CF

Security

EXAMPLE: PRESIGNED GET

```
myBucket = application.s3ServiceObject.bucket(“your
bucket name”);

presignedGETRequest  = {

    “duration”: “2h”,

    “key”: “fileName.txt”

};

awsResponse =
myBucket.generateGetPresignedUrl(presignedGETRequest);

signedURL = awsResponse.url;
```

Security

TIME-EXPIRING URLS IN CF2018 AND EARLIER

S3 SIGNING COMPONENT

```
signingUtils = CreateObject("component",
“s3RequestSigningUtils").init(accessKey, secretKey);

signedURL = signingUtils
.createSignedURL(s3BucketName, pathToFileInBucket);
```

github.com/brianklaas/ctlS3Utils

ENCRYPTING OBJECTS AT REST

ENCRYPTING OBJECTS AT REST

- **Server-Side Encryption with Amazon S3-Managed Keys** (SSE-S3)
- Server-Side Encryption with Customer Master Keys (CMKs) Stored in AWS Key Management Service (SSE-KMS)
- Server-Side Encryption with Customer-Provided Keys (SSE-C)

EXAMPLE: ENCRYPT OBJECTS AT REST

```
myBucket = application.s3ServiceObject.bucket("your bucket name");

uploadRequest  = {

    "srcFile": "path/to/local/file.txt",

    "key": "fileName.txt",

    "serverSideEncryption": "AES_256_SERVER_SIDE_ENCRYPTION"

};

uploadResponse = myBucket.uploadFile(uploadRequest);
```

Security

ENCRYPTING OBJECTS AT REST: CF2018 VERSION

```
fileContent = fileReadBinary(getTempDirectory() & uploadedFile.serverFile);

objectMetadata = CreateObject('java', 'com.amazonaws.services.s3.model.ObjectMetadata').init();
objectMetadata.setContentLength(ArrayLen(fileContent));
objectMetadata.setSSEAlgorithm(ObjectMetadata.AES_256_SERVER_SIDE_ENCRYPTION);

// Storing a file with Server-Side Encryption requires a byte stream, not a standard Java file object
fileInputStream = CreateObject('java', 'java.io.ByteArrayInputStream').init(fileContent);

putFileRequest = CreateObject('java', 'com.amazonaws.services.s3.model.PutObjectRequest')
    .init(s3BucketName, fileName, fileInputStream, objectMetadata);

s3.putObject(putFileRequest);
```

OBJECT LOCK

OBJECT LOCK

- Don’t allow files to be deleted
- Retention period = only for a period of time
- Legal Hold = forever (or until hold is removed)
- Enabled at the bucket level, overridden on a per-file basis

EXAMPLE: CONFIGURING OBJECT LOCK ON A FILE

```
myBucket = application.s3ServiceObject.bucket(“your bucket name”);

objectLockRequest  = {

    “key”: “fileName.txt”,

    “objectLockConfiguration”: {

        “objectLockEnabled”: “ENABLED”,

        “defaultRetention”: {

            “mode”: “COMPLIANCE”,

            “days”: 30

        }

    }

};

awsResponse = myBucket.putObjectLockConfiguration(objectLockRequest);
```

MONEY

STORAGE CLASSES

Money

STORAGE CLASSES

- S3 Standard
- S3 Intelligent-Tiering
- S3 Standard-Infrequent Access (S3 Standard-IA)
- S3 One Zone-Infrequent Access (S3 One Zone-IA)
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive

Money

STORAGE CLASSES

A DIGRESSION INTO DURABILITY

Money

STORAGE CLASSES: DURABILITY

- 11 9's of durability in ***distributed*** storage
- You're 400 times more likely to get hit by a meteor than lose one of a million objects in S3
- Formal proof-of-correctness algorithms
- Checksums (including bitslips in RAM)
- Actuarial models that anticipate when drives will fail
- Automated “durability auditors”

youtube.com/watch?v=nLyppihvhpQ

EXAMPLE: SET STORAGE CLASS

```
myBucket = application.s3ServiceObject.bucket("your bucket name");

uploadRequest  = {

    "srcFile": "path/to/local/file.txt",

    "key": "fileName.txt",

    "storageClass": "STANDARD_IA"

};

uploadResponse = myBucket.uploadFile(uploadRequest);
```

Money

STORAGE CLASSES

A DIGRESSION INTO COST

Money

STORAGE CLASSES: COST

S3 Standard Storage	
First 50TB/month	$0.023 per GB
S3 Intelligent - Tiering	
Frequent Access Tier, First 50 TB / Month	$0.023 per GB
Infrequent Access Tier, All Storage / Month	$0.0125 per GB
Monitoring and Automation, All Storage / Month	$0.0025 per 1,000 objects
S3 Standard - Infrequent Access	
All Storage / Month	$0.0125 per GB
S3 One Zone - Infrequent Access	
All Storage / Month	$0.01 per GB
S3 Glacier Flexible	
All Storage / Month	$0.0036 per GB
S3 Glacier Deep Archive	
All Storage / Month	$0.00099 per GB

Money

STORAGE CLASSES: COST

BEWARE EGRESS

Money

STORAGE CLASSES: COST

CONSIDER CLOUDFRONT

4.6TB + CloudFront = $406

4.6TB via S3 alone = $414

LIFECYCLE FLOWS

(Automatic Archiving)

Money

LIFECYCLE FLOWS

DON’T PAY FOR FILES YOU DON’T NEED

Money

LIFECYCLE FLOWS

Transition Actions:

"After 90 days, move all files to infrequent access storage."

Expiration Actions:

"After 180 days, delete the file."

Money

LIFECYCLE FLOWS

Pro Tips:

- Lifecycle rule to move files to 1ZIA after 30 days
  - Minimum of 30 days in standard storage required
  - 128KB minimum file size still enforced for IA storage classes
- You can only have one “all files” predicate per bucket
- Filter by object key (path) prefix, or tags

POWER

VERSIONING

Power

VERSIONING

- Whole bucket only
- On object upload, S3 returns the version ID in the version-id property of the response object
- **You** are responsible for keeping track of what versions mean

Security

EXAMPLE: GET VERSIONS OF AN OBJECT

```
myBucket = application.s3ServiceObject.bucket(“your bucket name”);

versionsRequest  = {

    “prefix”: “path/to/file.txt”

};

versions = myBucket.listAllVersions(versionsRequest);
```

Power

VERSIONING: CAUTION

- You pay for every version
- Older versions kept even when you suspend versioning
- Set lifecycle rule to have old versions auto-expire after [n] days

TAGS

Power

TAGS

FIND THINGS IN A HUMAN-READABLE WAY

TAGS

Business		Technical		Security	
Cost Center	41001	Environment	Dev	Compliance	HIPAA
Department	Security	Zone	Frontend	Data Sensitivity	4
Owner	Bill Bridges	Application	Order-Fullfilment	Confidentiality	Restricted

- Key-value pairs
- Up to 10 per object

EXAMPLE: ADDING TAGS TO OBJECTS

```
myBucket = application.s3ServiceObject.bucket(“your bucket name”);

addTagsRequest  = {

    “key”: “path/to/file.txt”,

    “tags” : [

        { “key”: “department”, “value”: “finance” }

        { “key”: “project”, “value”: “dashboards2022” }

    ]

};

awsResponse = myBucket.addTags(addTagsRequest);
```

Power

TAGS

Tagging Guides:

- https://aws.amazon.com/answers/account-management/aws-tagging-strategies/
- https://k9security.io/docs/guide-to-tagging-cloud-deployments/

TO INFINITY
...AND BEYOND

WEBSITE HOSTING

Great for static/JAMstack sites

S3 AS A DATABASE

Amazon Athena

S3 Select

S3 EVENTS

Lambda listeners

S3 Batch

MORE SECURITY

MFA on delete

Named access points

Attribute (or tag)-based access control (ABAC)

REPLICATION

Automatic cross-region replication

DataSync for on-prem data

REQUESTER PAYS

You don’t pay for egress!

GO DO!

Thank you!

Brian Klaas

brian.klaas@gmail.com

@brian_klaas

Blog: brianklaas.net

github.com/brianklaas/awsPlaybox