Might use IAM/bucket policy to lock down S3, but need identity of user to map it for access anyway …whether I can hardcode S3 credentials into app depends on whether I am ok keeping webpage on local laptop only! Once in S3 we need authentication

So…

Using this to setup S3 bucket and configure for using Javascript in the browser

<https://aws.amazon.com/developers/getting-started/browser/>

Facebook App Id created is: 754651064672675

Plncr (<https://plnkr.co/edit/?p=catalogue>) now allows me to see facebook login button using my Login.html code

Rest of logic to control login navigation

[**https://developers.facebook.com/docs/facebook-login/web/**](https://developers.facebook.com/docs/facebook-login/web/)

khphotoshowweb needs an index.html to host site (separate public-facing bucket)

Had to put the open CORS config in both PhotoShow and PhotoShowWeb buckets to get from Network Failure to Access Denied

ACTUALLY NOW WAS POINTING TO WRONG BUCKET IN JS – @TODO FIX LATER

…but am definitely past Facebook login

Used Bootstrap for styles as per Pluralsight training

Use mongoose to serve up files, as tiny web server

Remember Facebook App ID goes into trust relationship within IAM role, which in turn links to policy granting access to S3 bucket, so messing around with facebook Apps requires editing of AWS config. Otherwise you get “missing credentials in config” type error

# S3 Security

See use case #4: Restricting Access to a Specific HTTP Header

<https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html#example-bucket-policies-use-case-4>

….by default all S3 resources are private except to the fella who created it (i.e. me as Root IAM!)

The facebook example originally had a PutObjectAcl permission and used it to make all objects public:

ACL: 'public-read'

So, I definitely need a Bucket Policy. CORS is not for authenticated content <http://stackoverflow.com/questions/38486505/how-to-restrict-aws-s3-access-using-cors>

…and I’ve gone to effort of doing login with facebook so should be able to control access via the role of the user logging in….

|  |
| --- |
| {  "Version": "2012-10-17",  "Id": "S3PolicyId1",  "Statement": [  {  "Sid": "IPAllow",  "Effect": "Allow",  "Principal": {  "AWS": "arn:aws:iam::827454618391:role/PhotoShowRole"  },  "Action": "s3:GetObject",  "Resource": "arn:aws:s3:::khphotoshow/\*",  "Condition": {  "IpAddress": {  "aws:SourceIp": "89.100.190.74"  }  }  }  ]  } |

Second part alone worked when Principal was \*, but once I add PhotoShowRole it doesn’t?

Why doesn’t it see my GET request as being from the authenticated facebook role?

Because I need to pass the auth header and stop creating a dumb URL? Or should I use a signed URL? No that would be slow and feels unnecessary.

Because here’s what my GET request looks like

1. GET /khphotoshow/facebook-1324508804245182/DSC04922.JPG HTTP/1.1 Host: s3-eu-west-1.amazonaws.com Connection: keep-alive Accept: image/webp,image/\*,\*/\*;q=0.8 User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.93 Safari/537.36 Referer: http://localhost:8080/PhotoShow/ Accept-Encoding: gzip, deflate, sdch Accept-Language: en-US,en;q=0.8

So for now I extend the working one to allow from web S3, but know that spoofing IP address or (more likely) spoofing web address would allow people to see my photos. Big deal(?!) Well would like to get it using the authorization header. NEXT STEP FOR THIS MIGHT BE switching to a Javascript call for the GetObject (rather than simple URL), or find a way to inject the authorization header (answer is probably here <https://docs.aws.amazon.com/AmazonS3/latest/API/sigv4-auth-using-authorization-header.html> )

|  |
| --- |
| {  "Version": "2012-10-17",  "Id": "S3PolicyId1",  "Statement": [  {  "Sid": "IPAllow",  "Effect": "Allow",  "Principal": "\*",  "Action": "s3:GetObject",  "Resource": "arn:aws:s3:::khphotoshow/\*",  "Condition": {  "IpAddress": {  "aws:SourceIp": "89.100.190.74"  }  }  },  {  "Sid": "Allow get requests originating from my static web S3 bucket - the primary environment for website",  "Effect": "Allow",  "Principal": "\*",  "Action": "s3:GetObject",  "Resource": "arn:aws:s3:::khphotoshow/\*",  "Condition": {  "StringLike": {  "aws:Referer": [  "http://khphotoshowweb.s3-website-eu-west-1.amazonaws.com/\*"  ]  }  }  }  ]  } |

403 (Forbidden)

Errors like these…

GET https://s3-eu-west-1.amazonaws.com/khphotoshow/media/2016/Show\_H12016\_056.JPG 403 (Forbidden)

…even after emptying the cache are probably because the SOURCE IP ADDRESS is wrong. Am I at an airport or hotel rather than at home!?!?!?

18th Sept

Going to change policy to allow ANY authenticated Facebook accounts access to the photos. So, changing below…

|  |
| --- |
| {  "Version": "2012-10-17",  "Statement": [  {  "Action": [  "s3:GetObject~~",~~  ~~"s3:PutObject",~~  ~~"s3:PutObjectAcl~~"  ],  "Resource": [  ~~"arn:aws:s3:::khphotoshow/facebook-${graph.facebook.com:id}/\*"~~  "arn:aws:s3:::khphotoshow/media/\*"  ],  "Effect": "Allow"  },  {  "Action": [  "s3:ListBucket"  ],  "Resource": [  "arn:aws:s3:::khphotoshow"  ],  "Effect": "Allow",  "Condition": {  "StringEquals": {  ~~"s3:prefix": "facebook-${graph.facebook.com:id}"~~  "s3:prefix": "media"  }  }  }  ]  } |

## Dynamo DB

Building a serverless data lake

<http://www.slideshare.net/AmazonWebServices/building-a-serverless-data-lake-on-aws-technical-301>

…which links to blog Building & Maintaining metadata catalogue without servers

<https://blogs.aws.amazon.com/bigdata/post/Tx2YRX3Y16CVQFZ/Building-and-Maintaining-an-Amazon-S3-Metadata-Index-without-Servers>

DynamoDB Album structure:



* Primary key is name
* Ordering is by from\_date\_ms for album ordering on front page
* Type is in case we decide to do generic theming
* Devices is in case we upload photos covering same date range that don’t relate to the album e.g. photos taken from others while we were on holidays