Beyond Virtual Machines: Tapping into the AWS Universe from FileMaker

ITG06

Jesse Barnum
President, 360Works

FILEMAKER DEVCON 2018
AUGUST 6-9 | GRAPEVINE, TX

Jesse founded 360Works in 1996

Primary or original developer for most of our products, including MirrorSync, CloudMail, 360Deploy, Zulu, and SuperContainer

Certified in all versions of FileMaker

Frequent speaker at DevCon

Received FBA award for outstanding community contributions

Started using AWS in 2009

Moved all critical business services to AWS

What will we cover

If you would like to participate in our demo, open
fmp://jesse.local/Personnel Records Kinesis demo
(use devcon/devcon to login)

Interact directly with AWS Web Service APIs

- No plug-ins, no third party software

Signing AWS calls

- This uses new calc functions and curl options in FileMaker 16 or later
- JSON, CryptAuthCode, CryptDigest, HexEncode, SortValues

Examples of useful AWS APIs for FileMaker developers

What will we NOT cover

If you would like to participate in our demo, open
fmp://jesse.local/Personnel Records Kinesis demo
(use devcon/devcon to login)

Deployment of FileMaker Server on AWS

Deployment of FileMaker Cloud

How many AWS services are there?

How many AWS services are there?

131

How many AWS services are there?

131

I will be demonstrating 6 of them

- Kinesis
- SQS
- Transcribe
- S3
- Translate
- Athena

See the complete list at https://aws.amazon.com/products/

Demo Simple Queue Service (SQS)

AWS v4 signing process

Documented at https://docs.aws.amazon.com/general/latest/gr/sigv4_signing.html.
Request with no authorization header:

```
POST /669916120315/Test HTTP/1.1
host: sqs.us-east-1.amazonaws.com
User-Agent: FileMaker/17.0
Accept: */*
Accept-Encoding: deflate, gzip
Content-Length: 63
Content-Type: application/x-www-form-urlencoded

Version=2012-11-05&Action=SendMessage&MessageBody=Hello%20World
```

AWS v4 signing process

Add required headers for Amazon API calls. Include a SHA256 hash of the POST body:

```
POST /669916120315/Test HTTP/1.1
host: sqs.us-east-1.amazonaws.com
User-Agent: FileMaker/17.0
Accept: */*
Accept-Encoding: deflate, gzip
Content-Length: 63
x-amz-content-sha256:
9296c223212ec744b1c7d6a90153b16eb4f3075834d15bad91db6976573a7fb5
x-amz-date: 20180601T170040Z
Content-Type: application/x-www-form-urlencoded

Version=2012-11-05&Action=SendMessage&MessageBody=Hello%20World
```

AWS v4 signing process

Rearrange into canonical version of request. See documentation for precise definition of canonicalization.

POST
/669916120315/Test

content-length:63
host:sqs.us-east-1.amazonaws.com
x-amz-content-sha256:9296c223212ec744b1c7d6a90153b16eb4f3075834d15bad91db6976573a7fb5
x-amz-date:20180601T170040Z

content-length;host;x-amz-content-sha256;x-amz-date
9296c223212ec744b1c7d6a90153b16eb4f3075834d15bad91db6976573a7fb5

AWS v4 signing process

Now combine the algorithm name, timestamp, region, service name with a hash of the canonical request

AWS4-HMAC-SHA256
20180601T170040Z
20180601/us-east-1/sqs/aws4_request
e5407f8d720921ec8f30cff47696d52afb846457771fe2686984c493a9bb1db2

AWS v4 signing process

Now hash the result of the previous step with your secret key, current date, and service name. Combine with the name of the algorithm used, AWS access key (which is not a secret), region, service name, and list of signed headers.

AWS4-HMAC-SHA256 Credential=AKIAIJ5TDS35YGW4N2CA/20180601/us-east-1/sqs/aws4_request, SignedHeaders=content-length;host;x-amz-content-sha256;x-amz-date,
Signature=3359303c866ce8be51b3c84acbe7f0a0f85db017ae65712d087cc74da007e1b2

AWS v4 signing process

Include that as the Authorization header, along with the new headers added during the signing process, to get the final HTTP request:

```
POST /669916120315/Test HTTP/1.1
host: sqs.us-east-1.amazonaws.com
User-Agent: FileMaker/17.0
Accept: */*
Accept-Encoding: deflate, gzip
Content-Length: 63
x-amz-content-sha256: 9296c223212ec744b1c7d6a90153b16eb4f3075834d15bad91db6976573a7fb5
x-amz-date: 20180601T170040Z
Authorization:  AWS4-HMAC-SHA256 Credential=AKIAIJ5TDS35YGW4N2CA/20180601/us-east-1/sqs/aws4_request,
SignedHeaders=content-length;host;x-amz-content-sha256;x-amz-date,
Signature=3359303c866ce8be51b3c84acbe7f0a0f85db017ae65712d087cc74da007e1b2
Content-Type: application/x-www-form-urlencoded

Version=2012-11-05&Action=SendMessage&MessageBody=Hello%20World
```

Why the complicated signature?

Can be used without SSL

Immune to replay attacks (after 5 minutes)

Avoids sending secret keys over the wire

Immune to man-in-the-middle attacks (with or without SSL)

Avoids extra login HTTP request

Stateless requests simplify parallel processing

Demo

Homework assignment

Read about these AWS services:

- Lambda and API Gateway
- RDS, Aurora, DynamoDB, and RedShift (compatible with MirrorSync)
- Route 53
- Elastic Load Balancing (use for clustering FileMaker Servers with MirrorSync)
- CloudWatch
- SimpleDB
- AWS Directory Service (for external authentication with FileMaker Server)
- Comprehend
- Rekognition
- SNS
- SES
- AppStream

Session Updates

This session WILL NOT have updates.

www.filemaker.com/devcon/speaker_updates

(This is also listed in DevCon2Go)

Thank You!

Please remember to fill out the session evaluation at:

www.filemaker.com/devcon/evaluations

Q&A