Level Up Your CF Apps with Amazon Web Services

Brian Klaas

bklaas@jhu.edu

@brian_klaas

e you currently using any of the following AWS services in producti

Start the presentation to activate live content

If you see this message in presentation mode, install the add-in or get help at PollEv.com/app

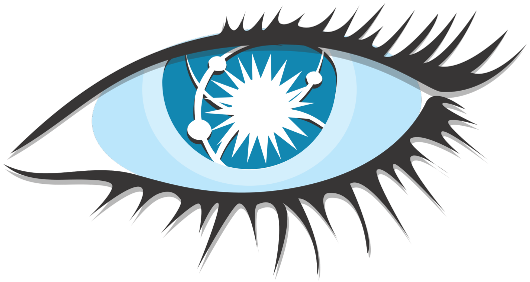
Level Up Your CF Apps with Amazon Web Services

Brian Klaas

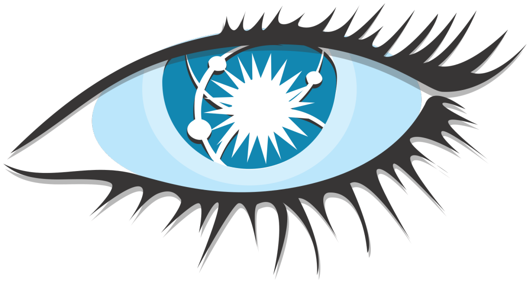
bklaas@jhu.edu

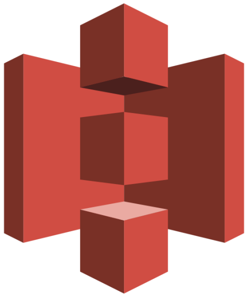
@brian_klaas

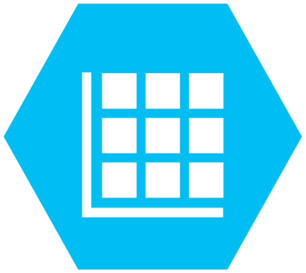
Hello



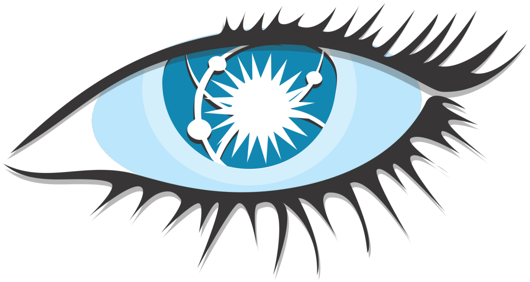
Hello

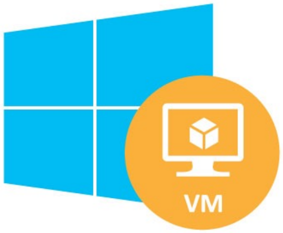




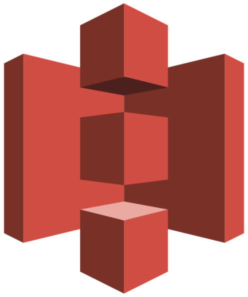


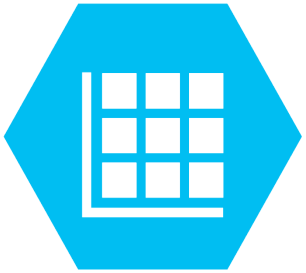
Hello





Hello





Hello

			Cloud Front	Glacier	S3	Storage Gateway	Snowball		
	Dynamo DB	Elasti Cache	RDS	Elastic Map Reduce	Redshift	Kinesis	Aurora	Athena	Elasticsearch
	Cloud Search	Elastic Transcoder	SES	SNS	SQS	SWF	FPS (Payment)	Mobile Analytics	Cognito
	App Stream	WorkMail and Docs	Work Spaces	AWS Config	VPC	IAM	Coud HSM	Route 53	Direct Connect
	Cloud Trail	Cloud Watch	Data Pipeline	Elastic Container Service	Batch	Lightsail	Xray	Code Build	Code Pipeline
Hello	EC2	Elastic Beanstalk	Lambda	Step Functions	API Gateway	Cloud Formation	OpsWorks	Code Commit	Code Deploy

Hello

Things you have to do on your own:

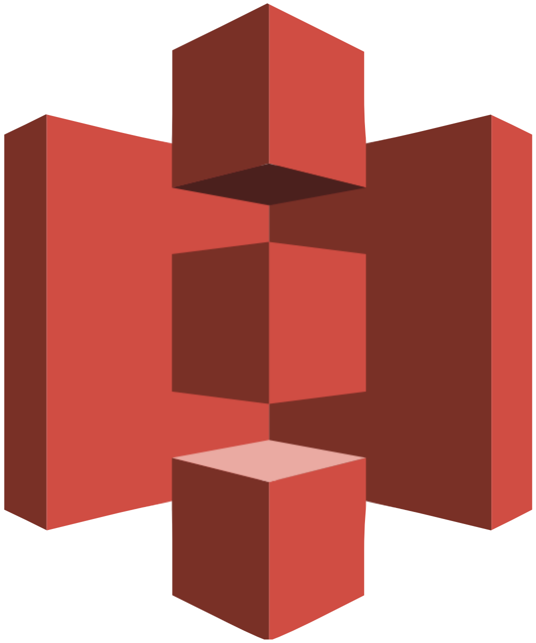
1. Go play in the console
2. Learn about IAM roles and permissions

Hello

AWS Playbox

github.com/brianklaas/awsplaybox

S3



Store all the things.

S3

99.999999999% durability

S3

$0.023 per GB stored

$0.004 per 10,000 GET

$0.005 per 1,000 PUT

$0.10 per GB out after 1GB

S3

Bucket

myfiles.s3.amazonaws.com

http://mybucket.s3.amazonaws.com/
path/to/file.png

S3

~~C:/~~

s3://

S3

```
cffile( variable="fileData", file="s3://
somebucket/somefile.txt", action="read" );
```

```
cfdirectory( directory="s3://somebucket/
someDirectory", action="list" );
```

S3

```
if ( !directoryExists("s3://
somebucket.s3.amazonaws.com") ) {
    perms = [ {group="all", permission="read"},
    {id="canonicalIDofYourAWSAccount",
     permission=“full_control"} ];
    cfdirectory( directory="s3://
somebucket.s3.amazonaws.com", storeacl=perms,
action="create" );
}

fileWrite("s3://somebucket.s3.amazonaws.com/
myFile.txt", "#someOutput#");

files = directoryList("s3://
somebucket.s3.amazonaws.com");
```

S3

IAM Credentials Required

S3

Inline:

```
cffile( variable="fileData", file="s3://
accessKey:secretKey@somebucket/somefile.txt",
action="read" );
```

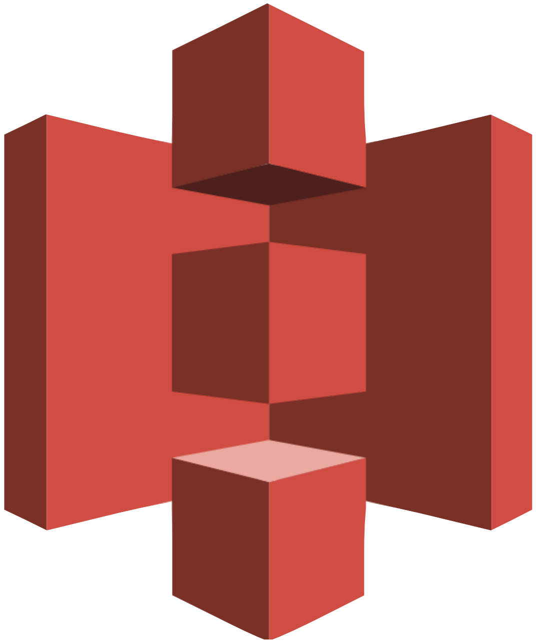
Or, in application.cfc:

```
this.s3.accessKeyId="accessKey";
this.s3.awsSecretKey="secretKey";
```

- Expire URLs
- Change properties on a per-request basis
- Upload to S3 from browser

Requires request signing.

S3



S3 Request Signing Utils

github.com/brianklaas/ctlS3utils

S3

S3 is storage, not a file system.

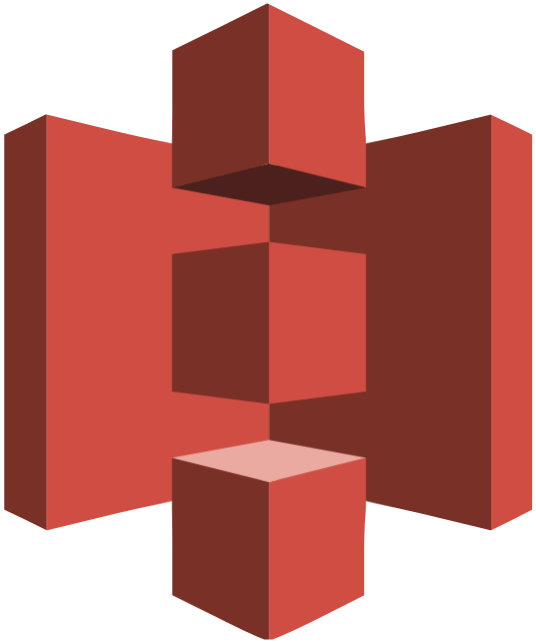
Can get basic file info with

```
cfhttp( url="http://bucket.s3.amazonaws.com/filename",
                method="head" );
```

CF only supports V2 signatures

Can’t use built-in CF support for S3 in new regions since 2014
(Frankfurt, Seoul, Mumbai, Ohio, Canada, London)

S3



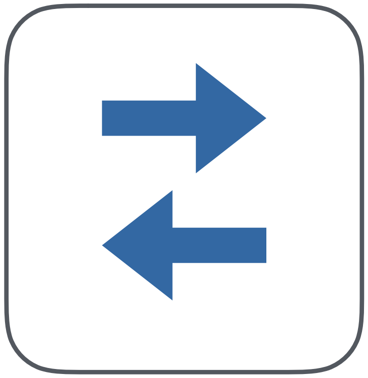
Store all the things.

S3

DynamoDB

= Hugely scalable,
high-write throughput
document data store

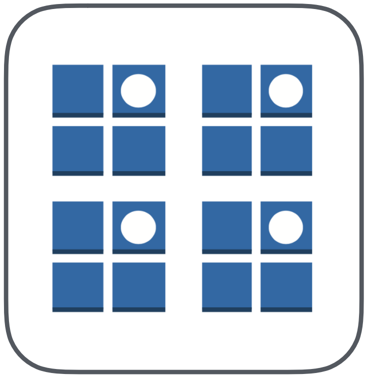
Dynamo
DB



1. Set read/write capacity

2. Set primary and sort keys

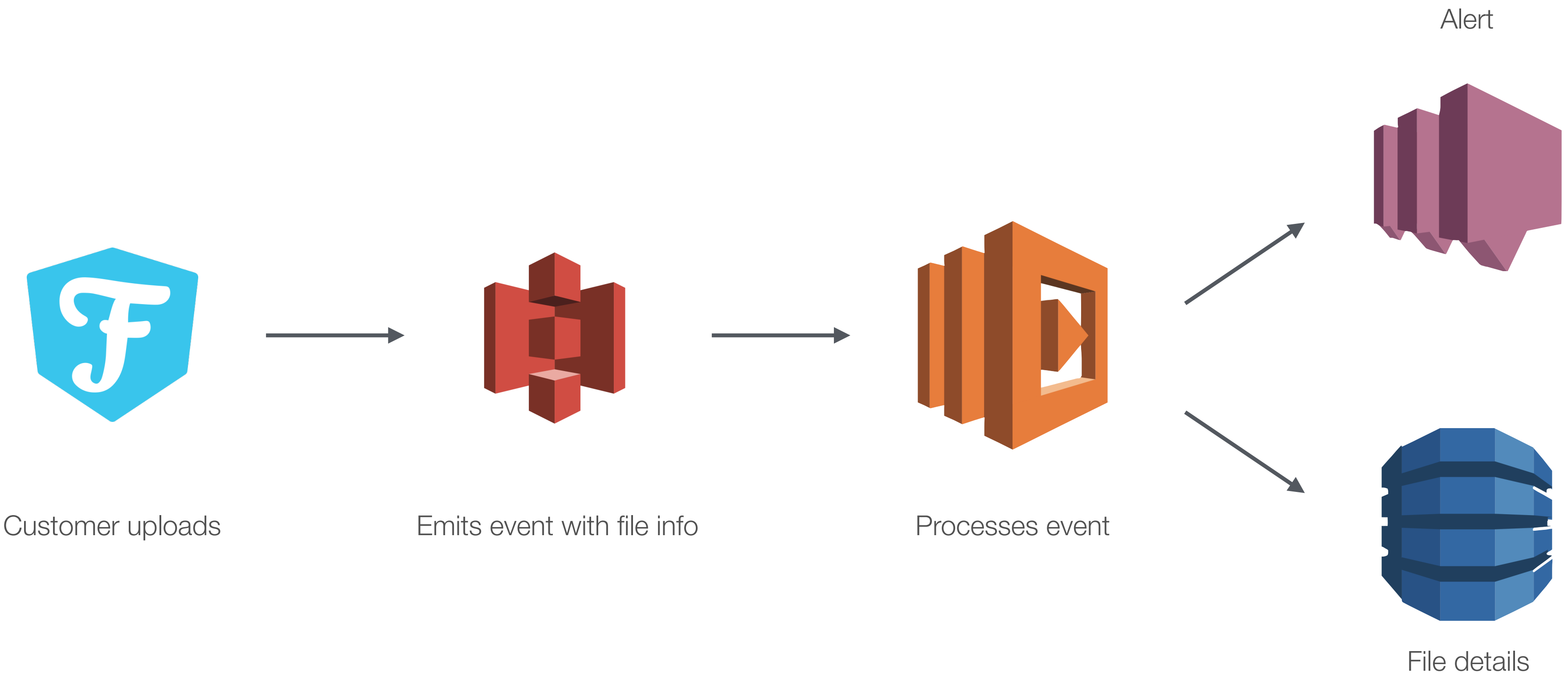
3. Set secondary indexes



4. Write

Dynamo
DB

Logging and Monitoring Third–Party Services



Alert

Customer uploads

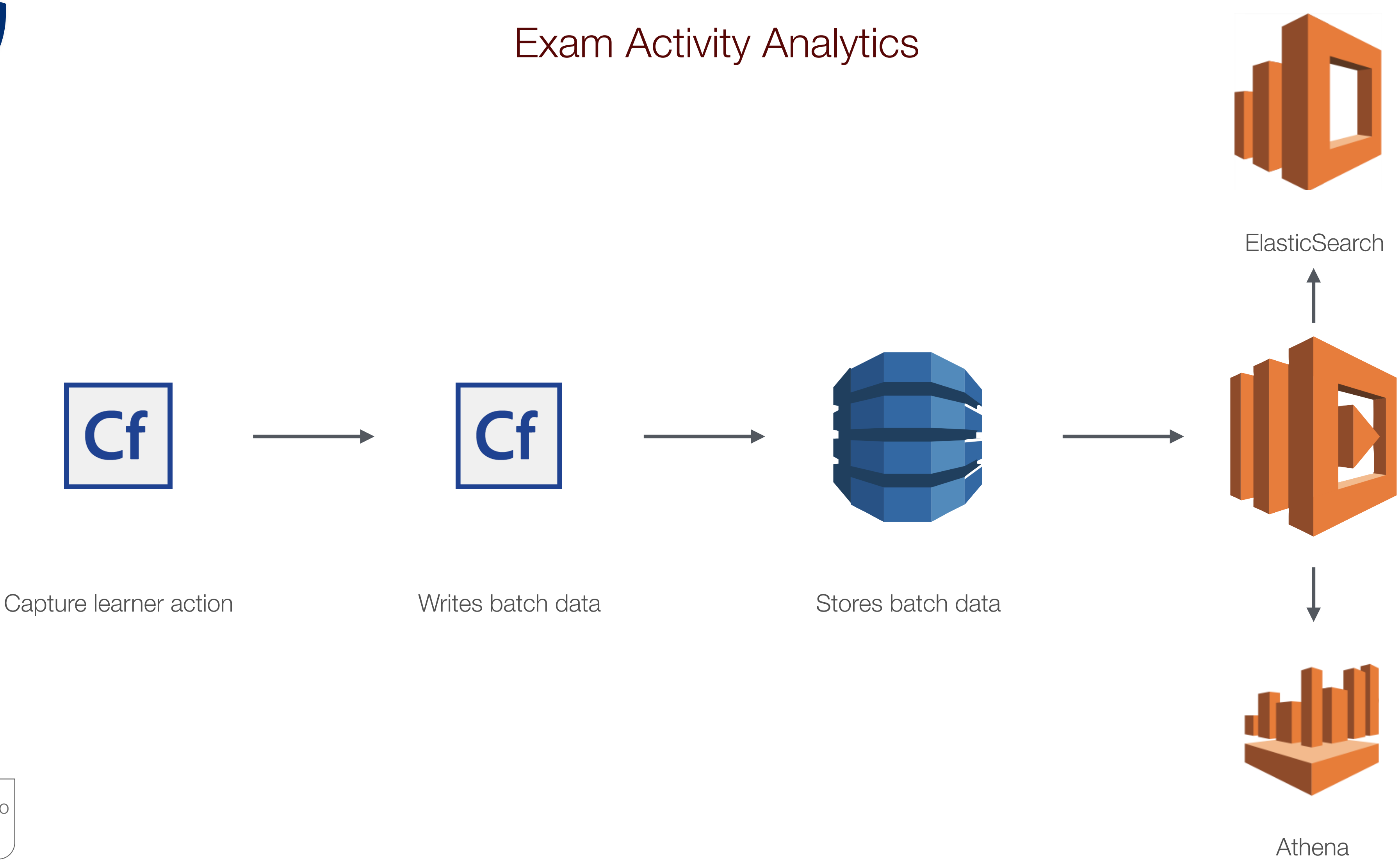
Emits event with file info

Processes event

File details

Dynamo
DB

Exam Activity Analytics



ElasticSearch

Capture learner action

Writes batch data

Stores batch data

Athena

Dynamo DB

Demo

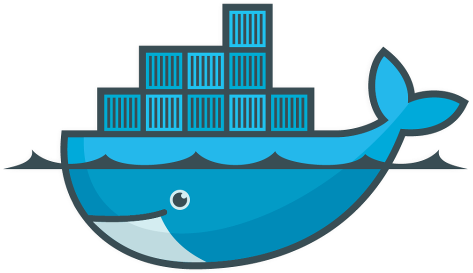
List, Put, Scan, Filter from Cf

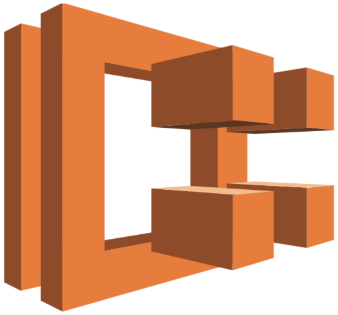
github.com/brianklaas/awsplaybox

Dynamo DB

Lambda

Lambda





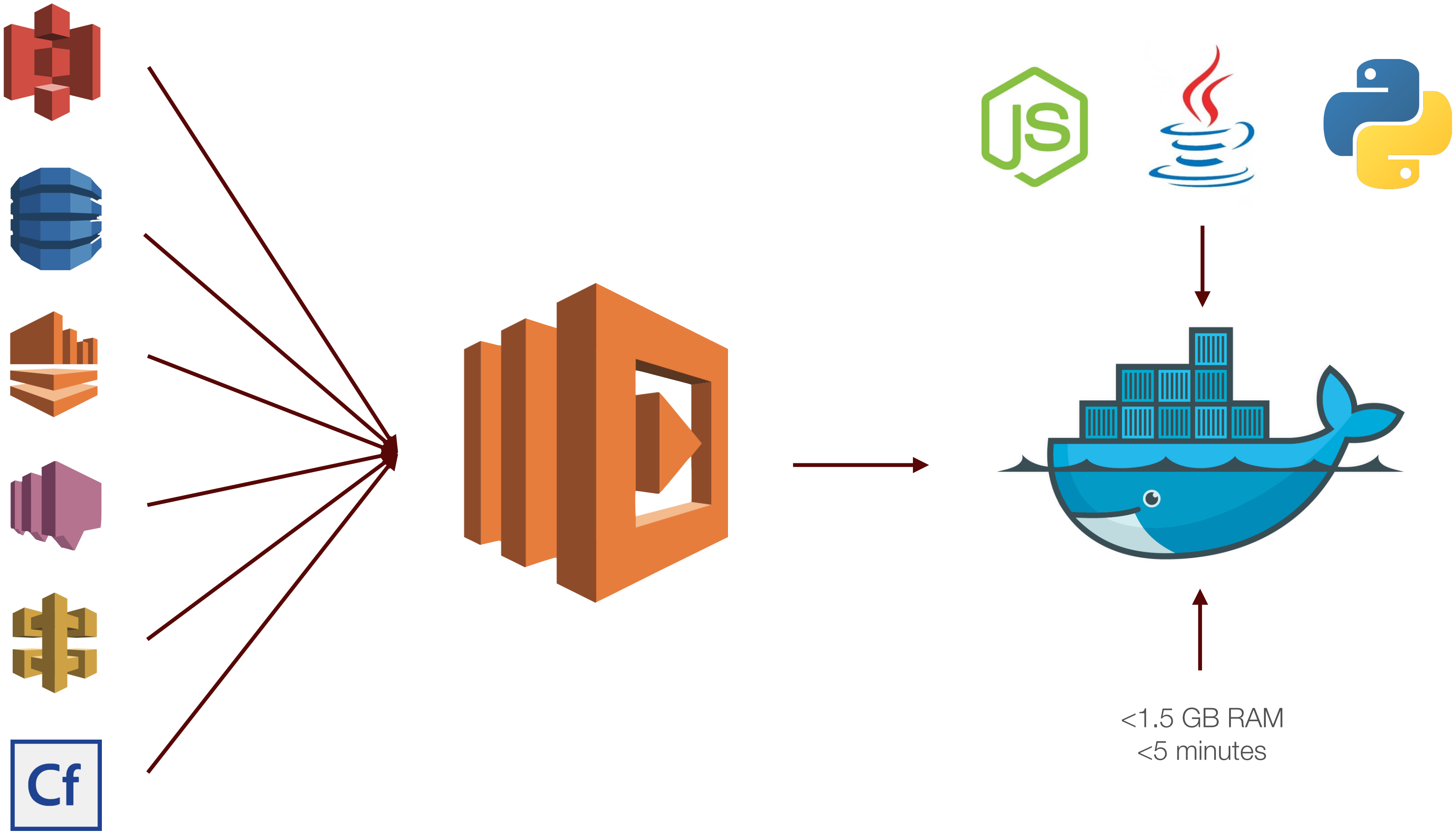
Lambda

No server is easier to manage than no server.

Lambda

= Event–driven computing

Lambda



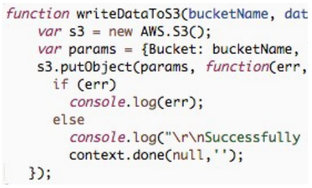
<1.5 GB RAM
<5 minutes

Lambda

1. Write a handler function

2. Upload ZIP

3. Invoke an event



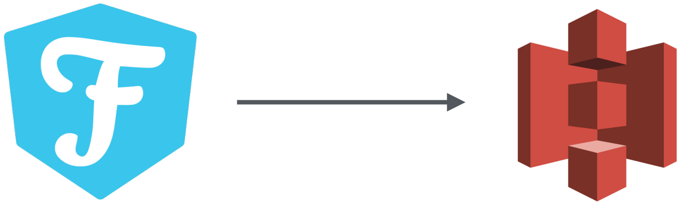
```
function writeDataToS3(bucketName, dat
    var s3 = new AWS.S3();
    var params = {Bucket: bucketName,
    s3.putObject(params, function(err,
      if (err)
        console.log(err);
      else
        console.log("\r\nSuccessfully
        context.done(null,'');
   });
```

4. Handler runs

= Microservices infrastructure without having to worry about running containers or scaling your infrastructure!

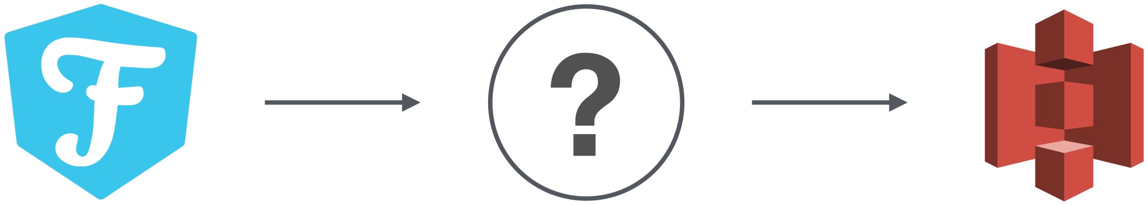
Lambda

Logging and Monitoring Third–Party Services



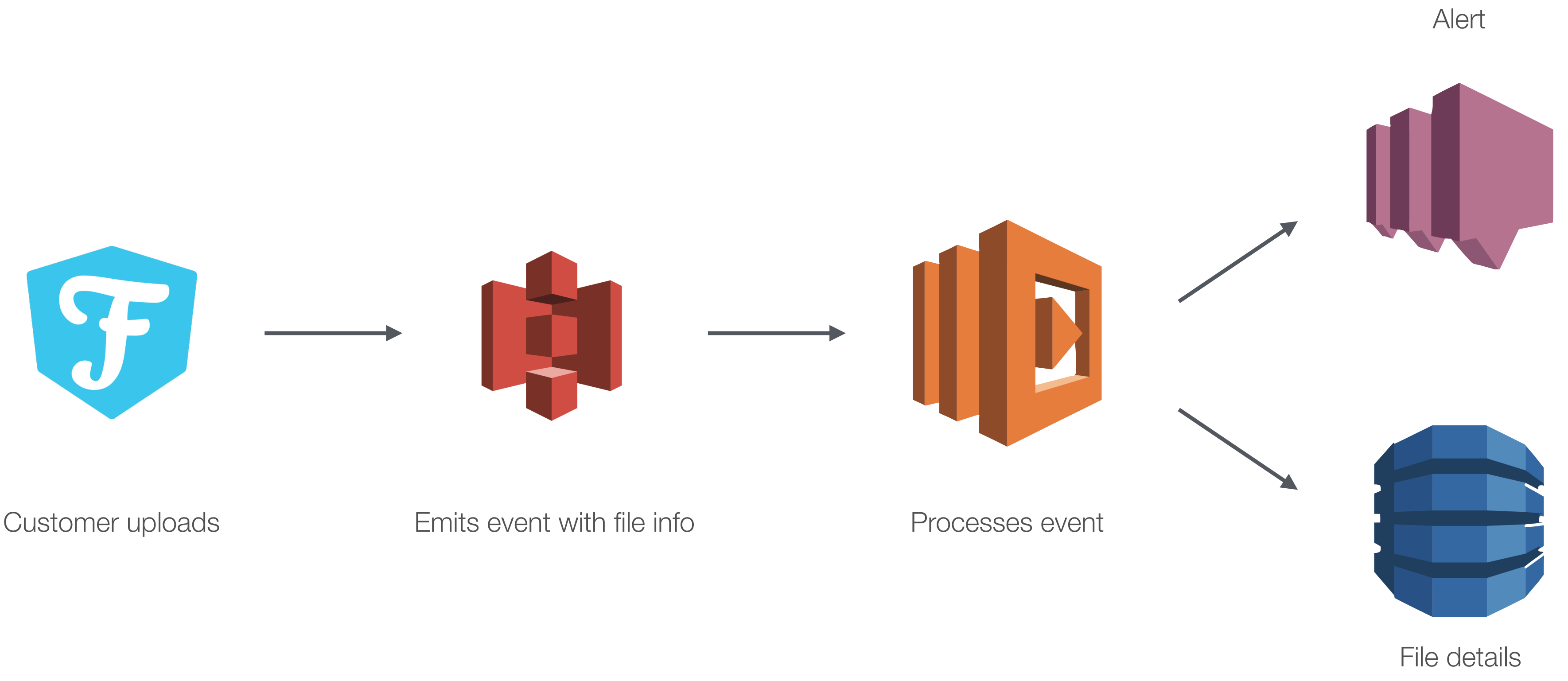
Lambda

Logging and Monitoring Third–Party Services



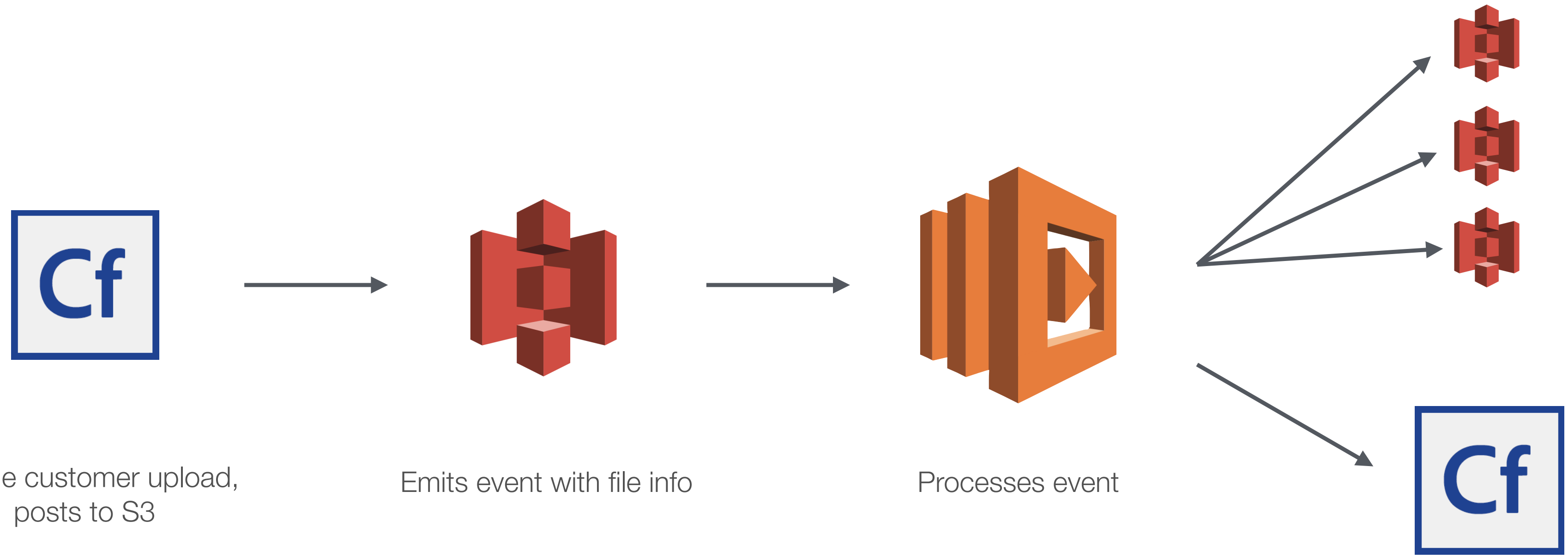
Lambda

Logging and Monitoring Third–Party Services



Lambda

Async Image Resizing



Multiple sizes written to S3

Handle customer upload, posts to S3

cffile (action="write" file="s3://somebucket/images.zip")

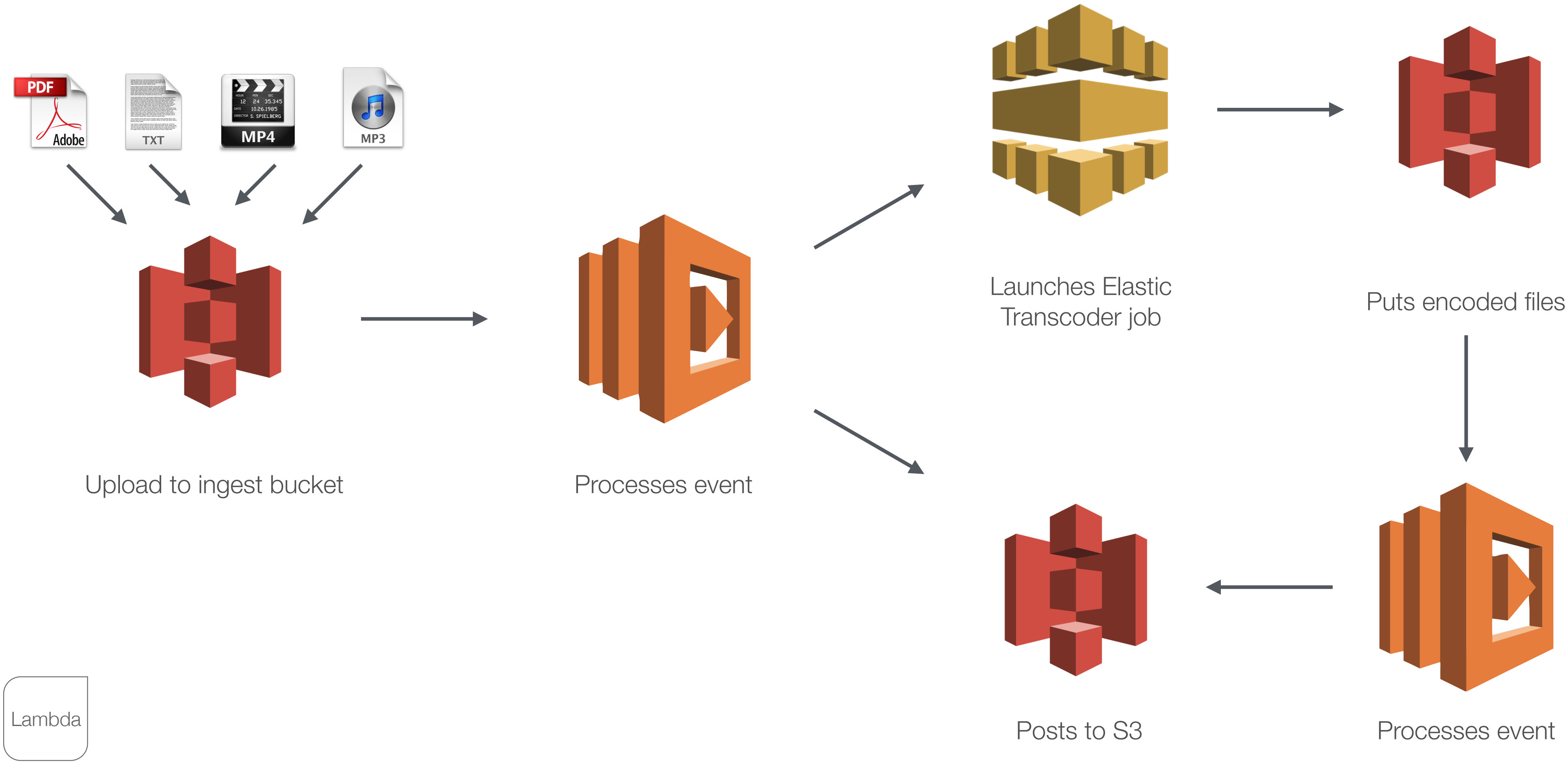
Emits event with file info

Processes event

Calls "done" http endpoint

Lambda

Video Production Workflow



Upload to ingest bucket

Processes event

Launches Elastic Transcoder job

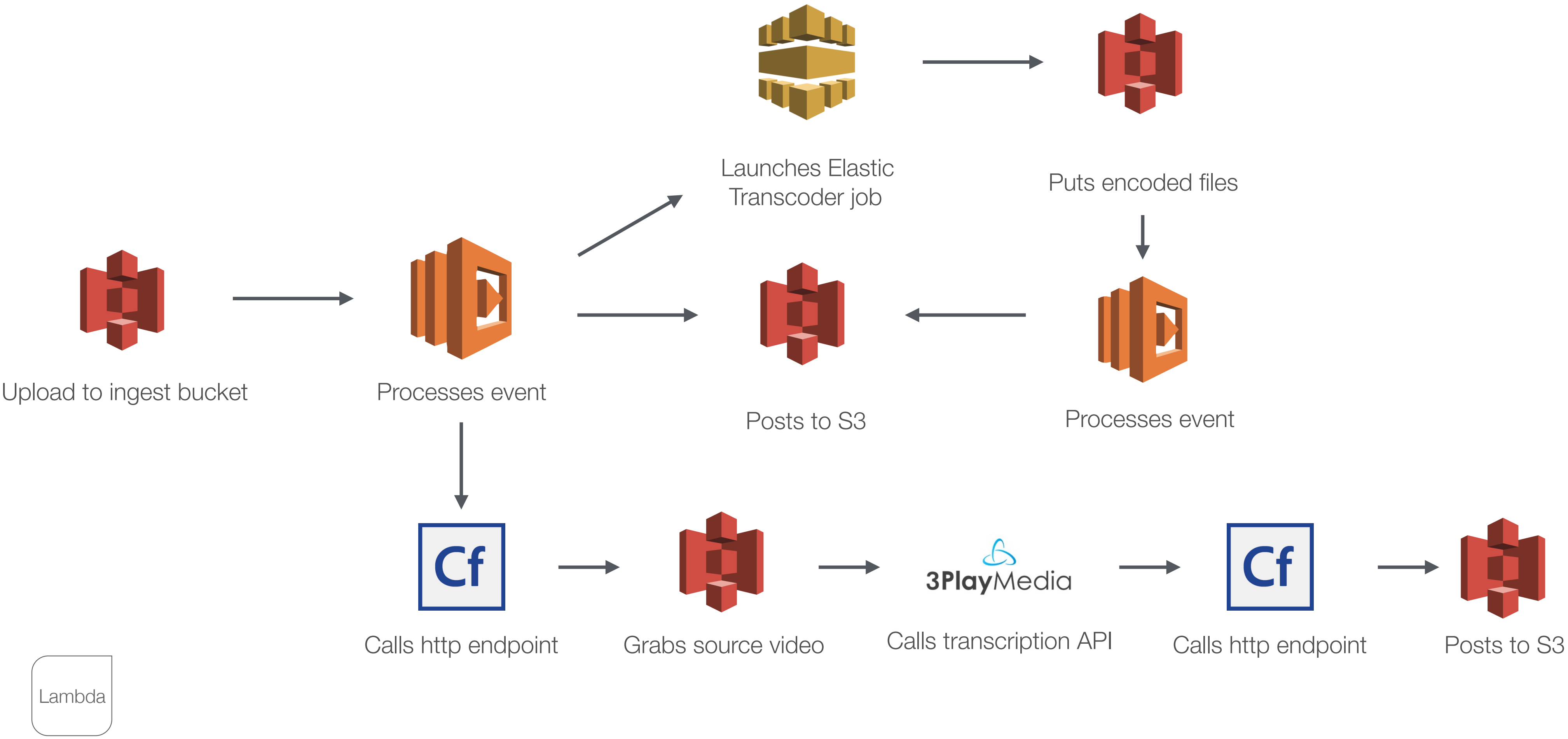
Puts encoded files

Processes event

Posts to S3

Lambda

Video Production Workflow (v2)



Upload to ingest bucket

Processes event

Launches Elastic
Transcoder job

Puts encoded files

Processes event

Posts to S3

Calls http endpoint

Grabs source video

Calls transcription API

Calls http endpoint

Posts to S3

Lambda

Demo

Invoking Lambda functions from Cf

github.com/brianklaas/awsplaybox

= Focus on building apps,
not infrastructure

Step Functions

(with some Rekognition)

Visual serverless orchestration

Step
Functions

Automated, multi-step, asynchronous, serverless workflows in AWS

Step Functions

Avoid rebuilding the monolith

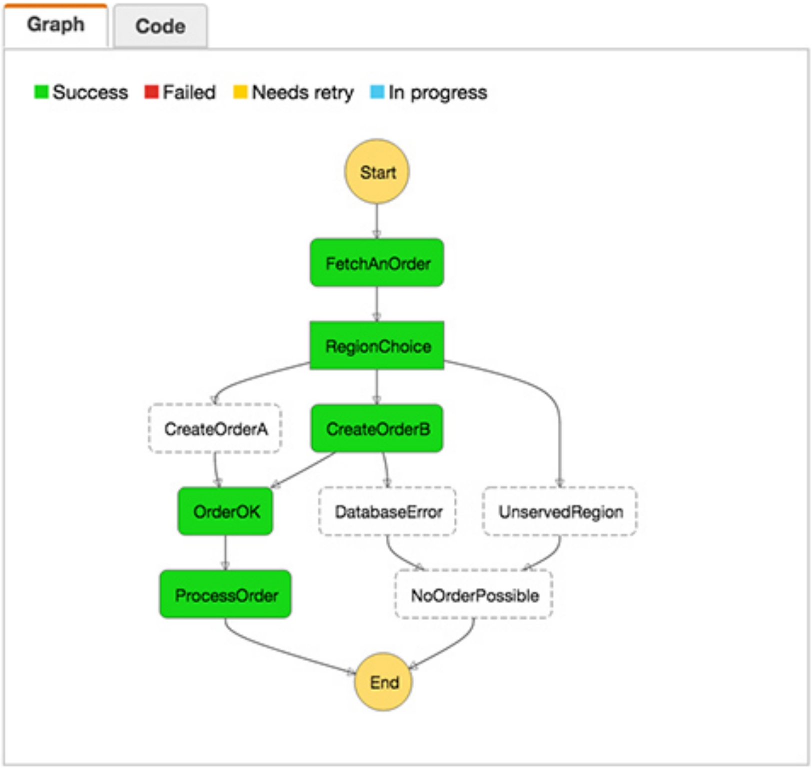
(Or the microlith)

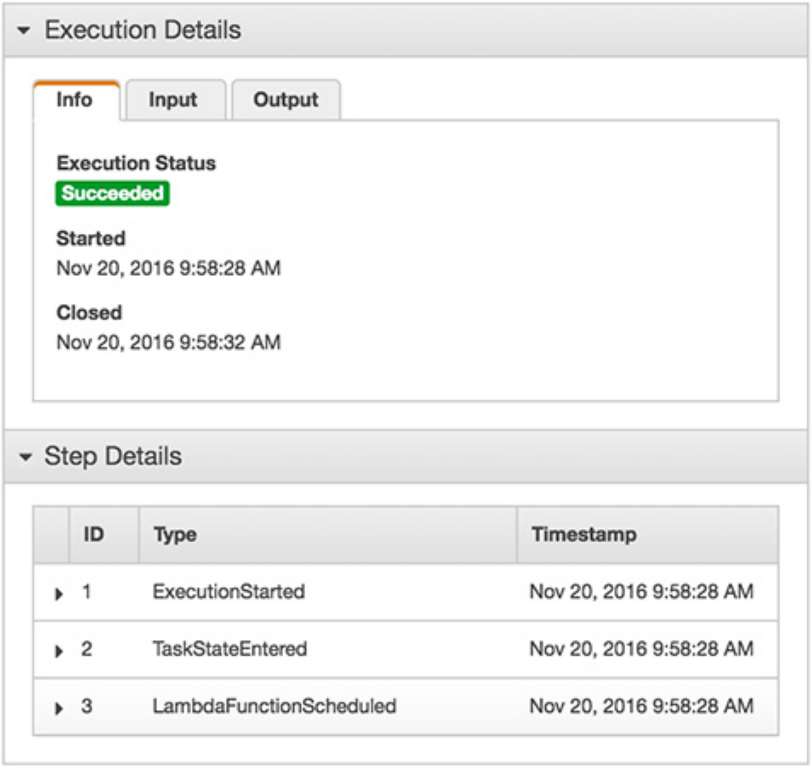
Step Functions

Step Functions = “state machine”

Amazon States Language (JSON)
https://states-language.net/spec.html

Step
Functions





Execution Details

Execution Status
Succeeded

Started
Nov 20, 2016 9:58:28 AM

Closed
Nov 20, 2016 9:58:32 AM

Step Details

	ID	Type	Timestamp
▸	1	ExecutionStarted	Nov 20, 2016 9:58:28 AM
▸	2	TaskStateEntered	Nov 20, 2016 9:58:28 AM
▸	3	LambdaFunctionScheduled	Nov 20, 2016 9:58:28 AM

Step Functions

Task

Sequence

Parallel

Branching

Waiting

Error Handling

Retries

Step
Functions

Demo

Invoking Step Functions from Cf

github.com/brianklaas/awsplaybox

Step
Functions

More on Rekognition in the awsPlaybox

Step
Functions

Automated, multi-step, asynchronous, serverless workflows in AWS

Step
Functions

Use statelint to validate your ASL JSON

https://github.com/awslabs/statelint

Step
Functions

Preparing for Outages

The Great S3 Outage of 2017

Plan for outages.

(Blame Amazon)

Batch upload from your servers

Use multiple regions

Shut off services

Outages

Have a plan.

Outages

Go Do!

AWS Playbox

github.com/brianklaas/awsplaybox

Please fill out
the session survey!

brian.klaas@gmail.com
@brian_klaas