Level Up Your Web Apps with Amazon Web Services

Brian Klaas

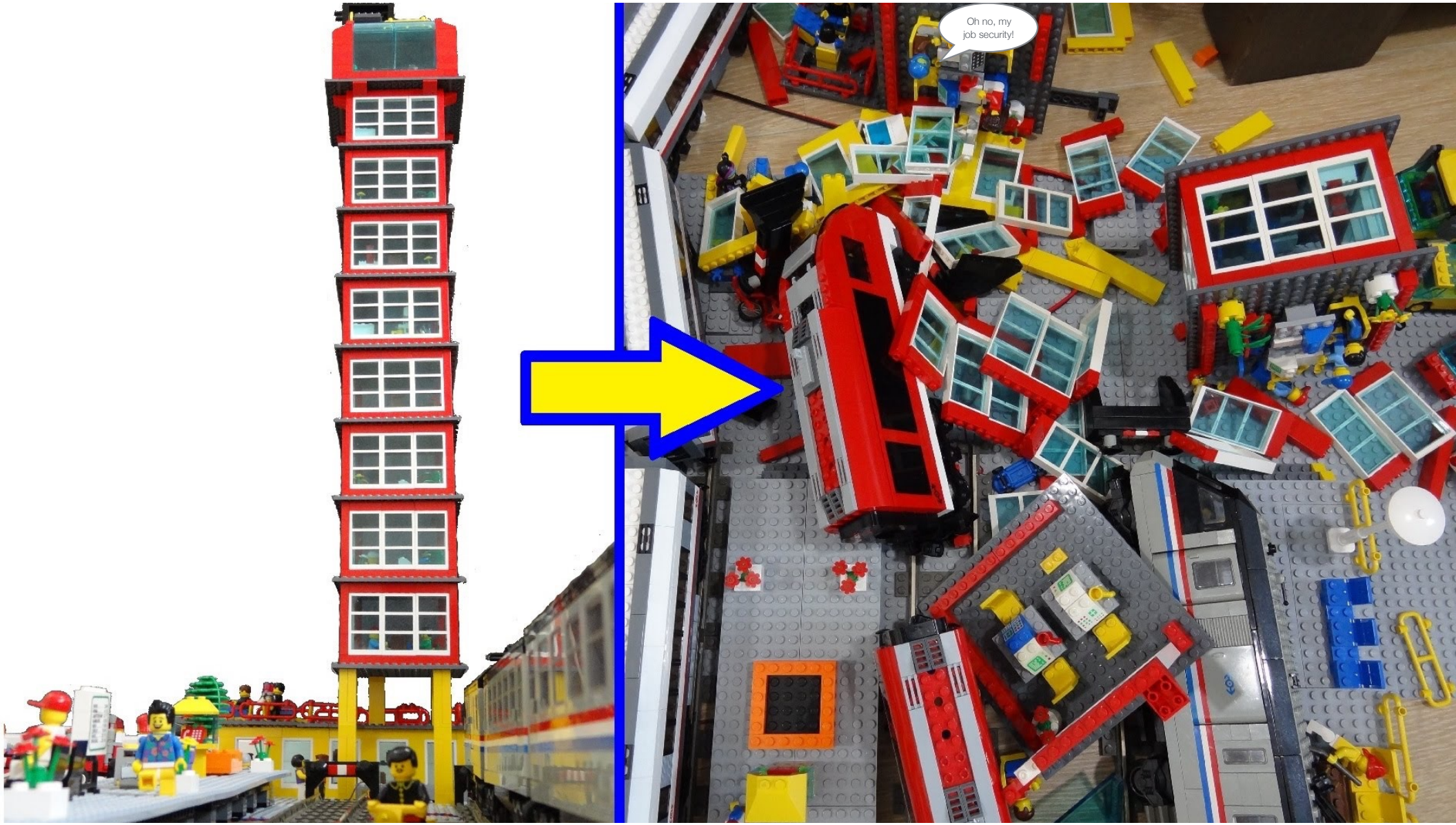
bklaas@jhu.edu

@brian_klaas

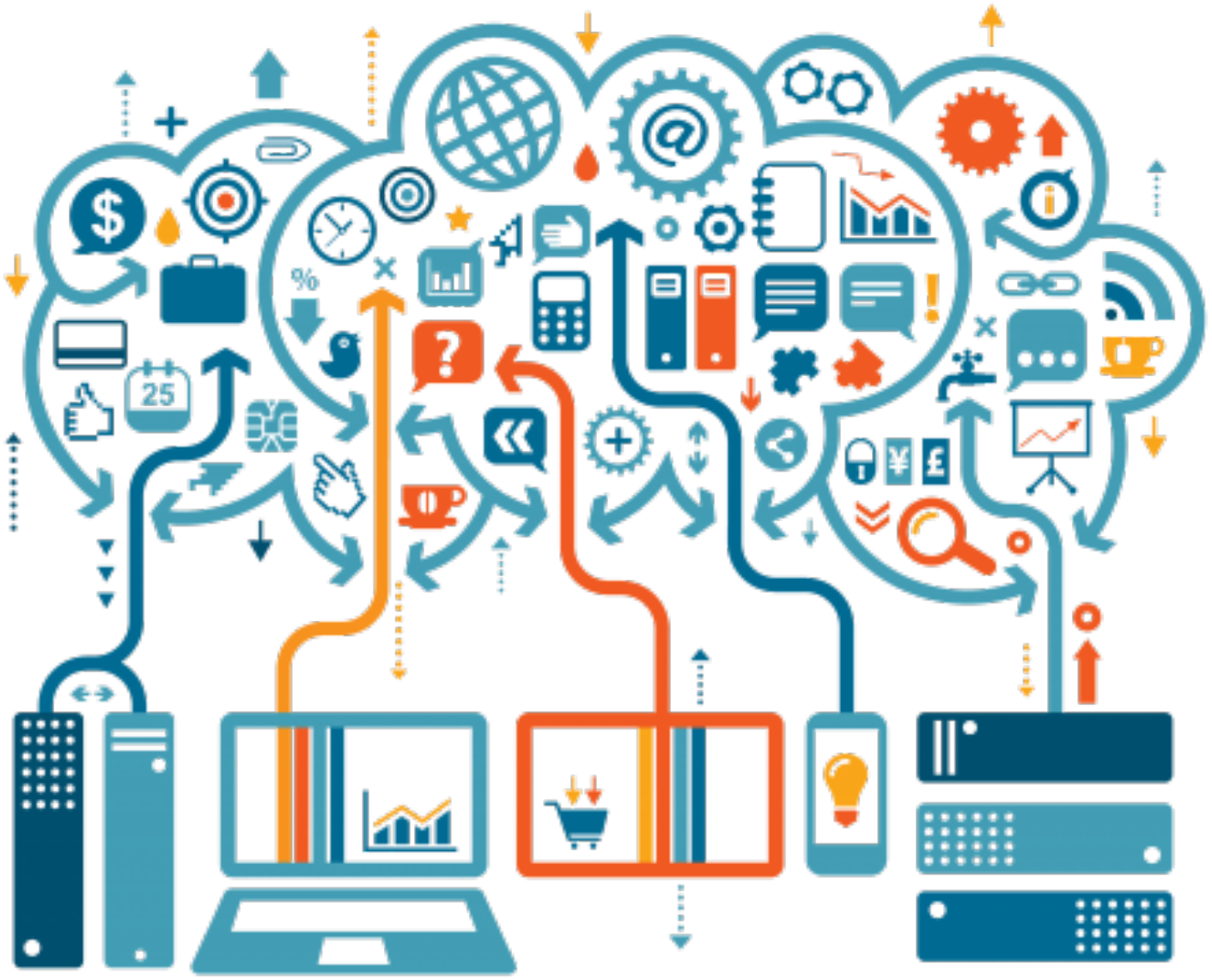
Hello



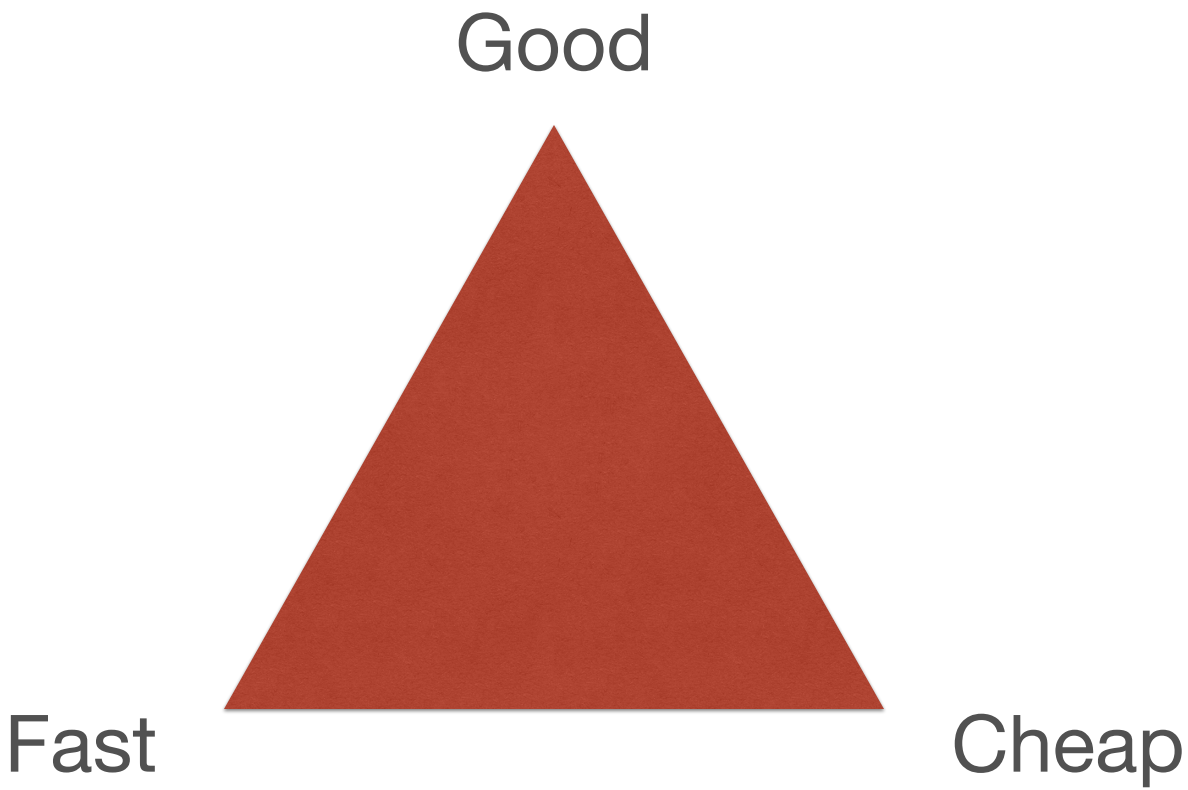
Hello


Oh no, my
job security!

Hello

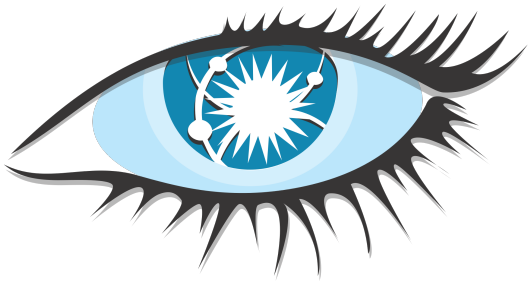


Hello

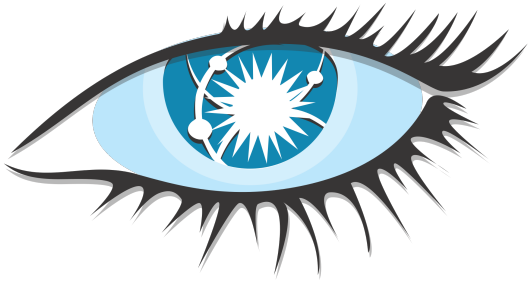


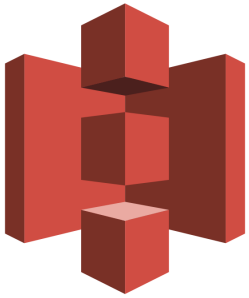
Hello

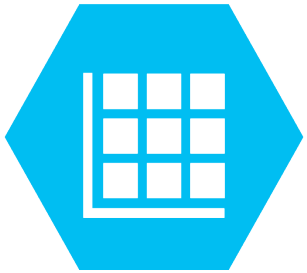
Hello



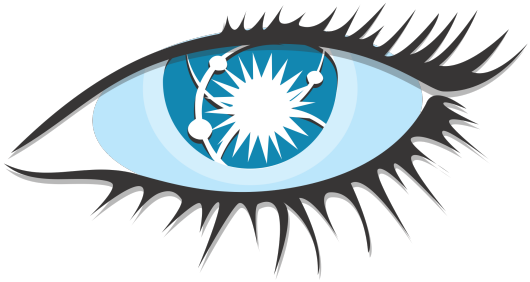
Hello

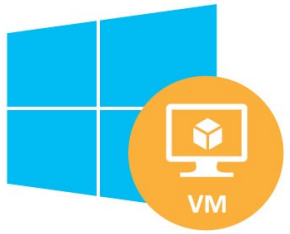


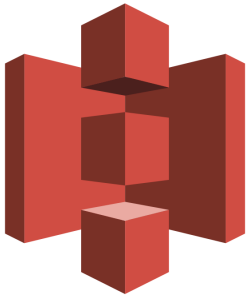


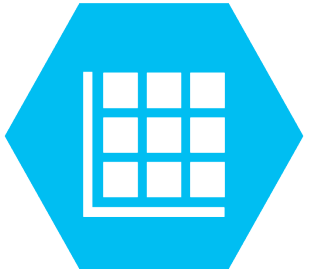


Hello



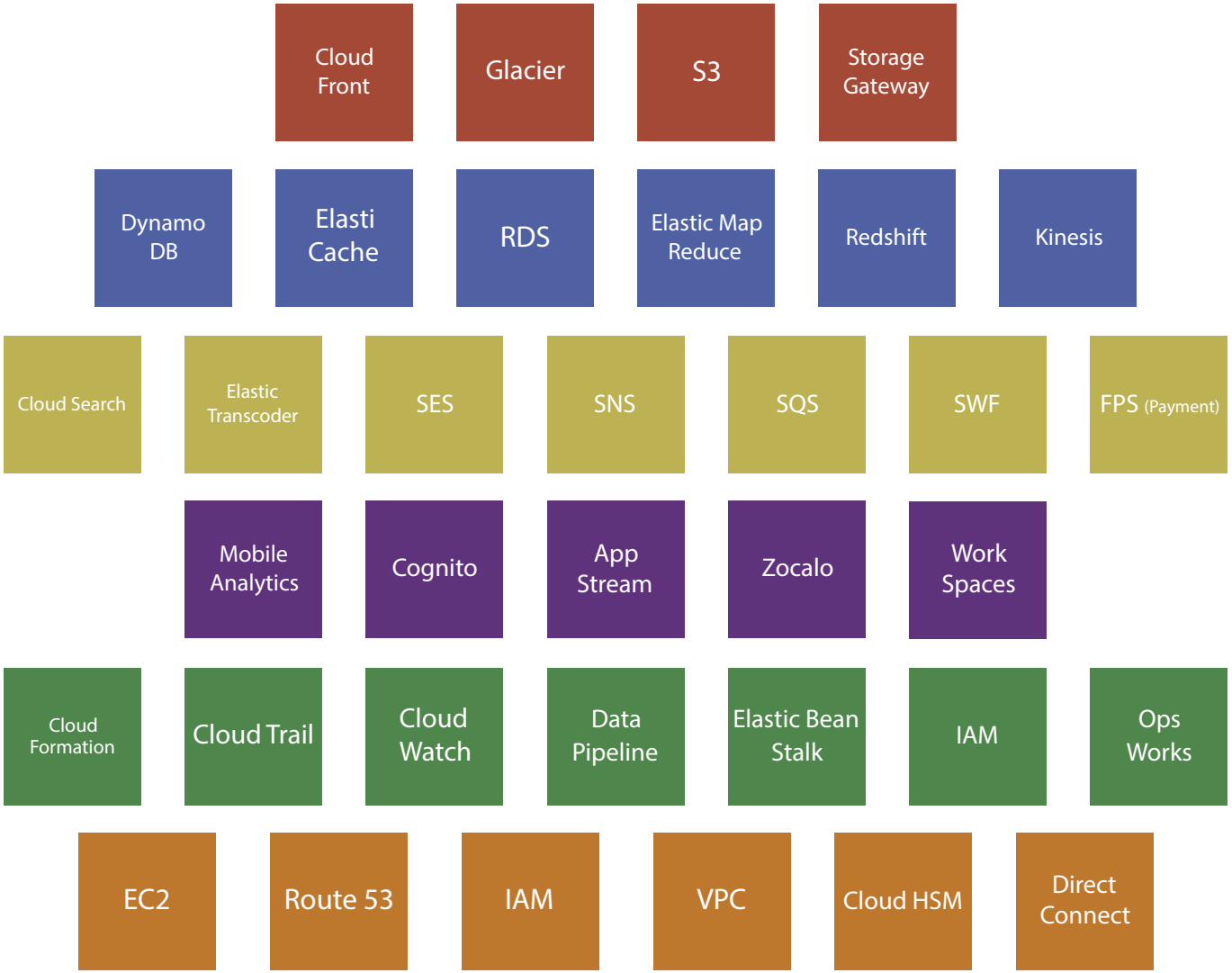






Hello

Hello

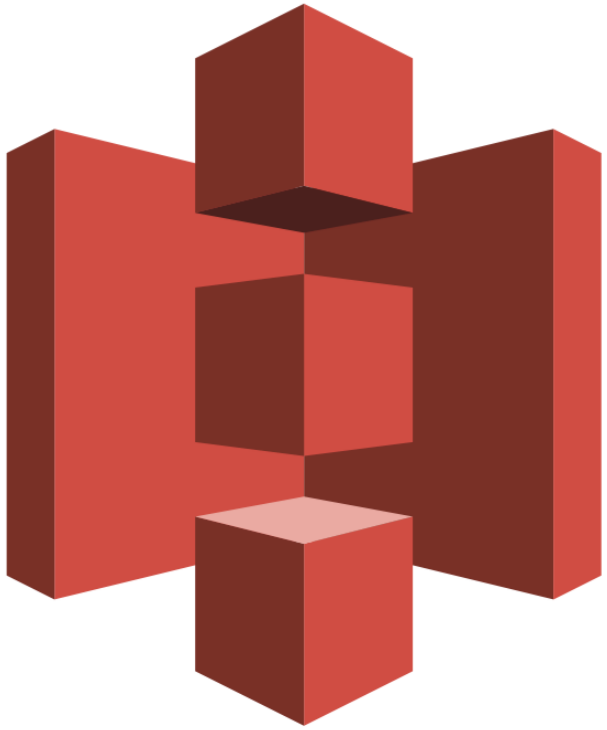

Cloud Front
Glacier
S3
Storage Gateway
Dynamo DB
Elasti Cache
RDS
Elastic Map Reduce
Redshift
Kinesis
Cloud Search
Elastic Transcoder
SES
SNS
SQS
SWF
FPS (Payment)
Mobile Analytics
Cognito
App Stream
Zocalo
Work Spaces
Cloud Formation
Cloud Trail
Cloud Watch
Data Pipeline
Elastic Bean Stalk
IAM
Ops Works
EC2
Route 53
IAM
VPC
Cloud HSM
Direct Connect

Hello

```
<cffile action="read"
file="s3://somebucket/somefile.txt"
variable="fileData" />
```

```
<cfdirectory action="list"
directory="s3://somebucket/someDirectory" />
```

Hello



S3

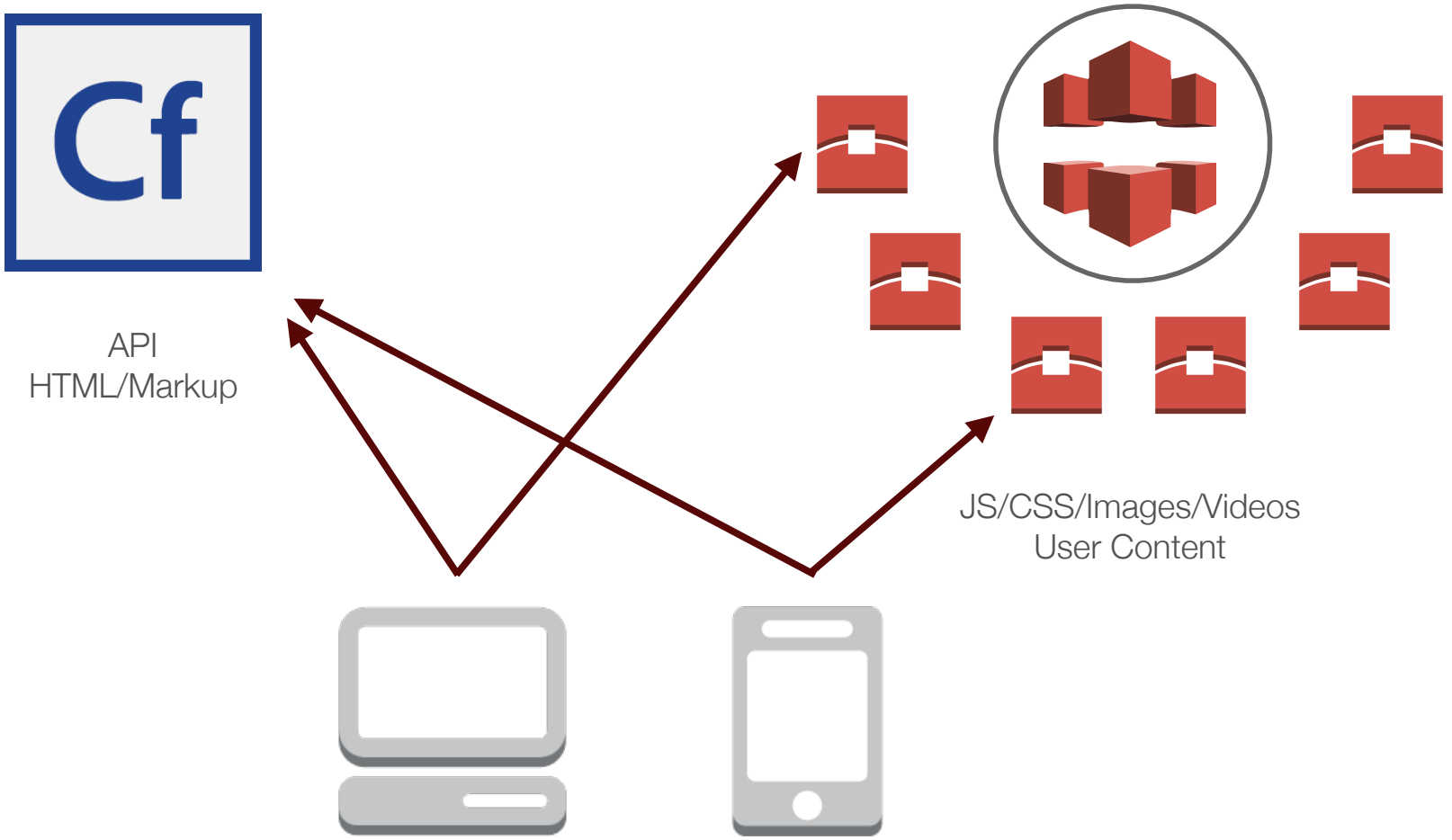
Hello

Hello

CloudFront


Greenland
Iceland
Finland
Sweden
Norway
Russia
Canada
United States
Mexico
North Atlantic Ocean
Poland
Ukraine
Kazakhstan
Mongolia
China
South Korea
North Pacific Ocean
Italy
Spain
Turkey
Iraq
Iran
Afghanistan
Pakistan
Algeria
Libya
Egypt
Saudi Arabia
Mali
Niger
Sudan
Chad
Nigeria
Ethiopia
Kenya
DR Congo
Tanzania
Angola
Namibia
Botswana
Madagascar
South Africa
Thailand
Indonesia
Papua New Guinea
Australia
New Zealand
Indian Ocean
Venezuela
Colombia
Peru
Brazil
Bolivia
Chile
Argentina
South Pacific Ocean
South Atlantic Ocean
Southern Ocean

Cloud Front



Cloud
Front

Progressive Download

Streaming

HTTP Post

Cloud Front

- Geographic distribution of video

  April 2016:

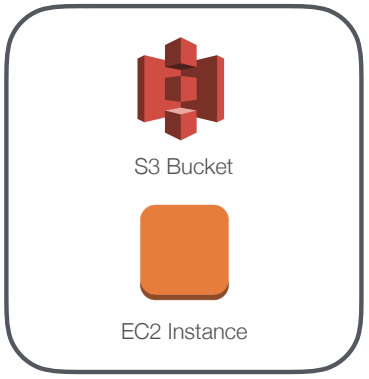
- Faster downloads

  $148.96

- Less traffic on our network

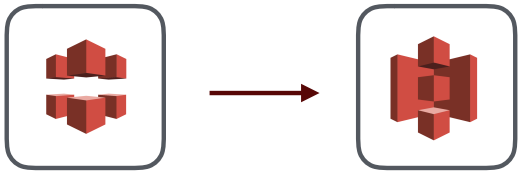
- ~3TB transfer/month

Cloud Front

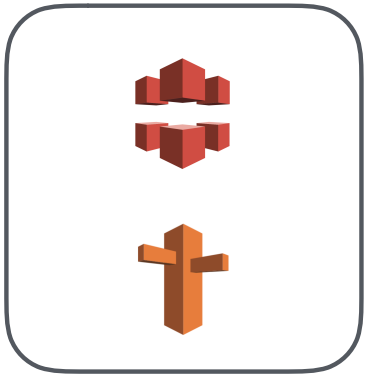


1. Set up source

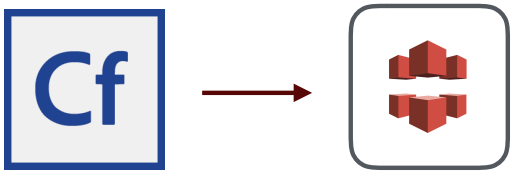
2. Create a distribution



3. Point distribution to source



4. Use generated DNS entry,
or your own



5. Point to CloudFront URLs
in your code

Cloud
Front

Congratulations!

Cloud
Front

Require CloudFront

Origin Access Identity

Requires signed requests

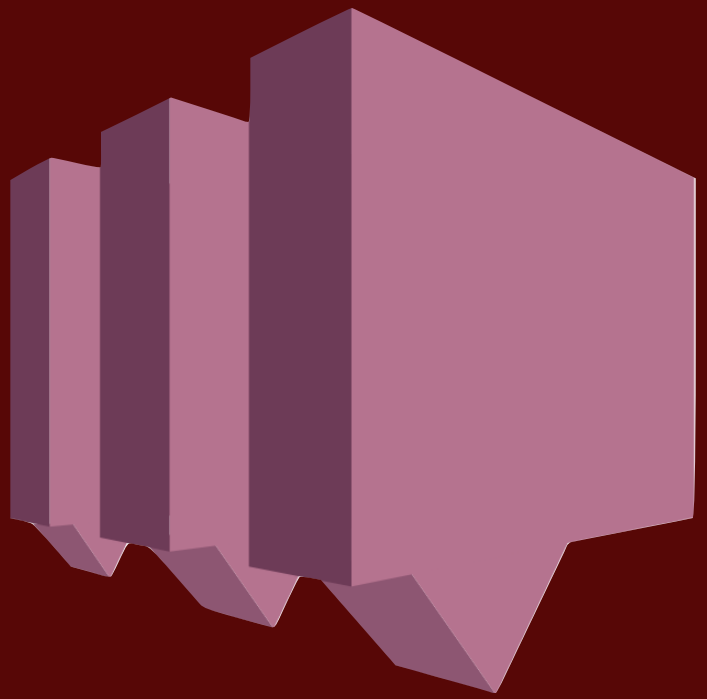
Versioning

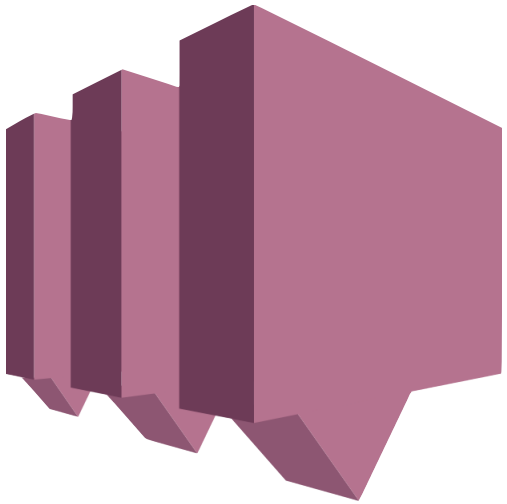
1. Unique names on every change via preprocessor
2. Sign request with new version number

CTL CloudFront Utils

https://github.com/brianklaas/ctlCloudFrontUtils

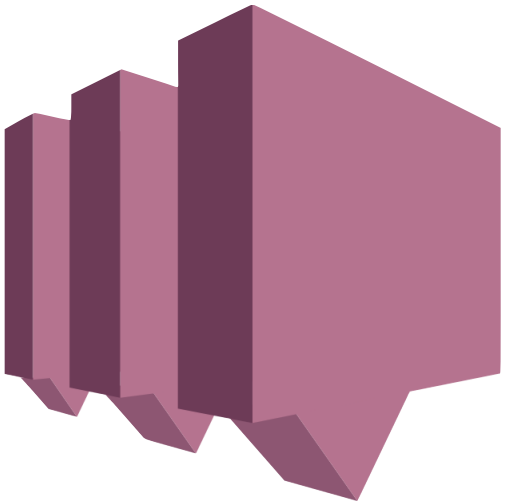
SNS





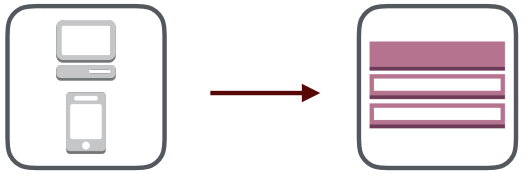
- Email
- SMS (US only)
- iOS/Android notifications
- HTTP/S endpoint
- Other AWS services

SNS

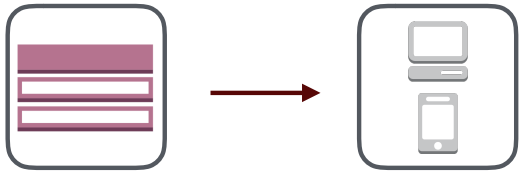


= Pub/Sub

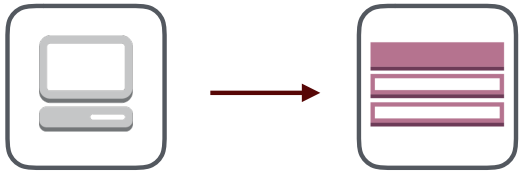
1. Create topic



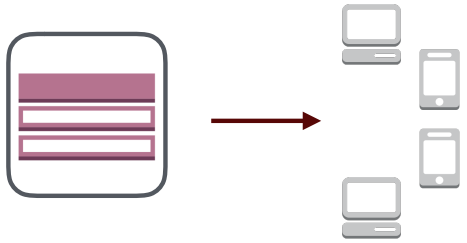
2. Authenticated request to create subscription



3. Consumer must confirm subscription



4. Authenticated request to create message



5. All consumers on topic receive message

SNS

Demo

Sending messages to a SNS topic from 

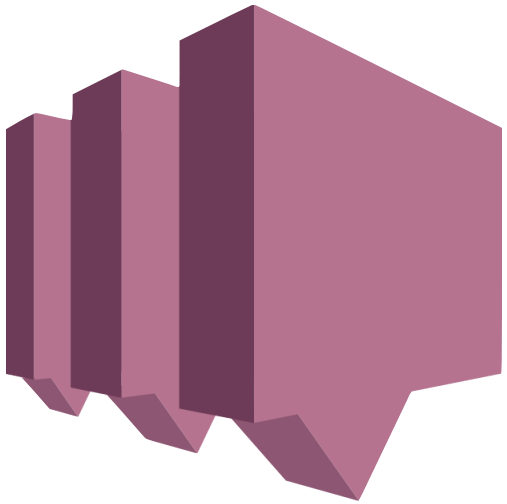
github.com/brianklaas

Things you have to do on your own:

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

Install pip + boto

1. Pip
   https://pip.pypa.io/en/stable/installing/
2. Boto
   http://boto.cloudhackers.com/en/latest/getting_started.html
3. Set up credentials for boto
   http://boto.cloudhackers.com/en/latest/boto_config_tut.html

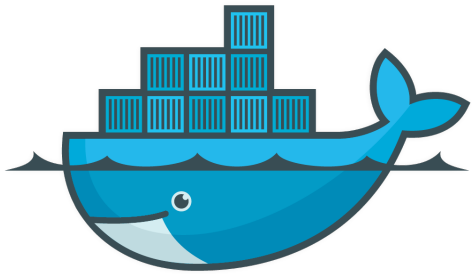


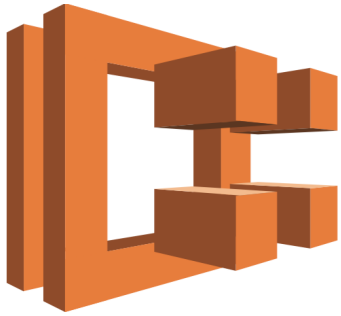
- Email
- SMS (US only)
- iOS/Android notifications via the Amazon Mobile SDK
- HTTP/S endpoint
- SQS, Lambda

SNS

Lambda

Lambda



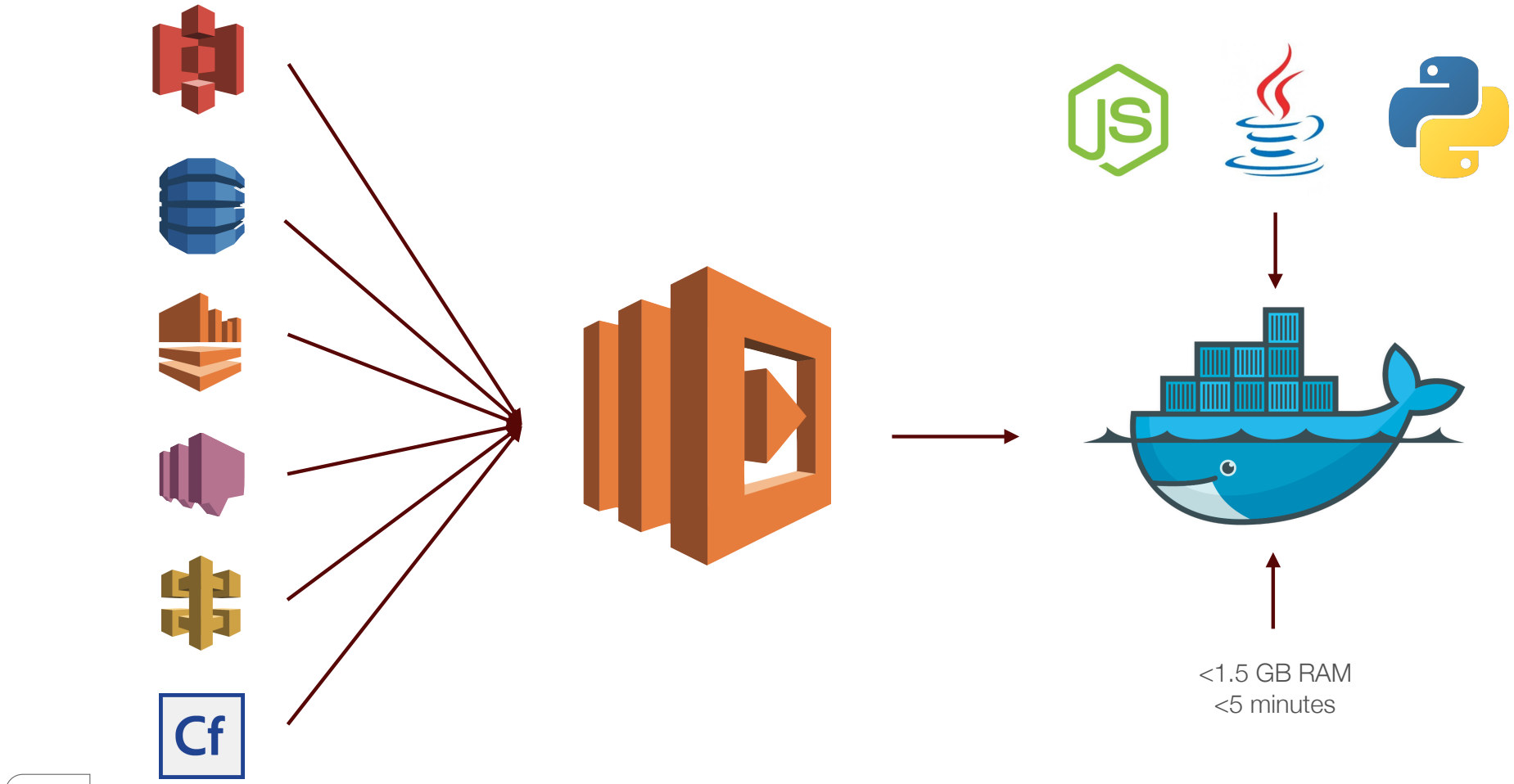


Lambda

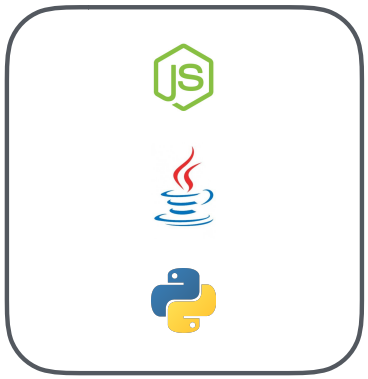
No server is easier to manage than no server.

= Event–driven computing

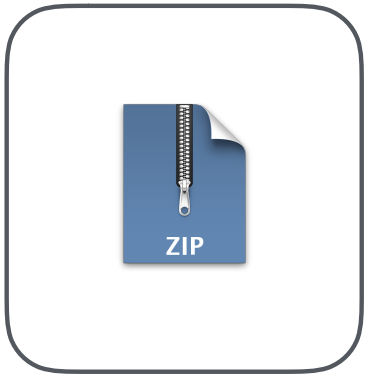
Lambda



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

Lambda

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

Lambda

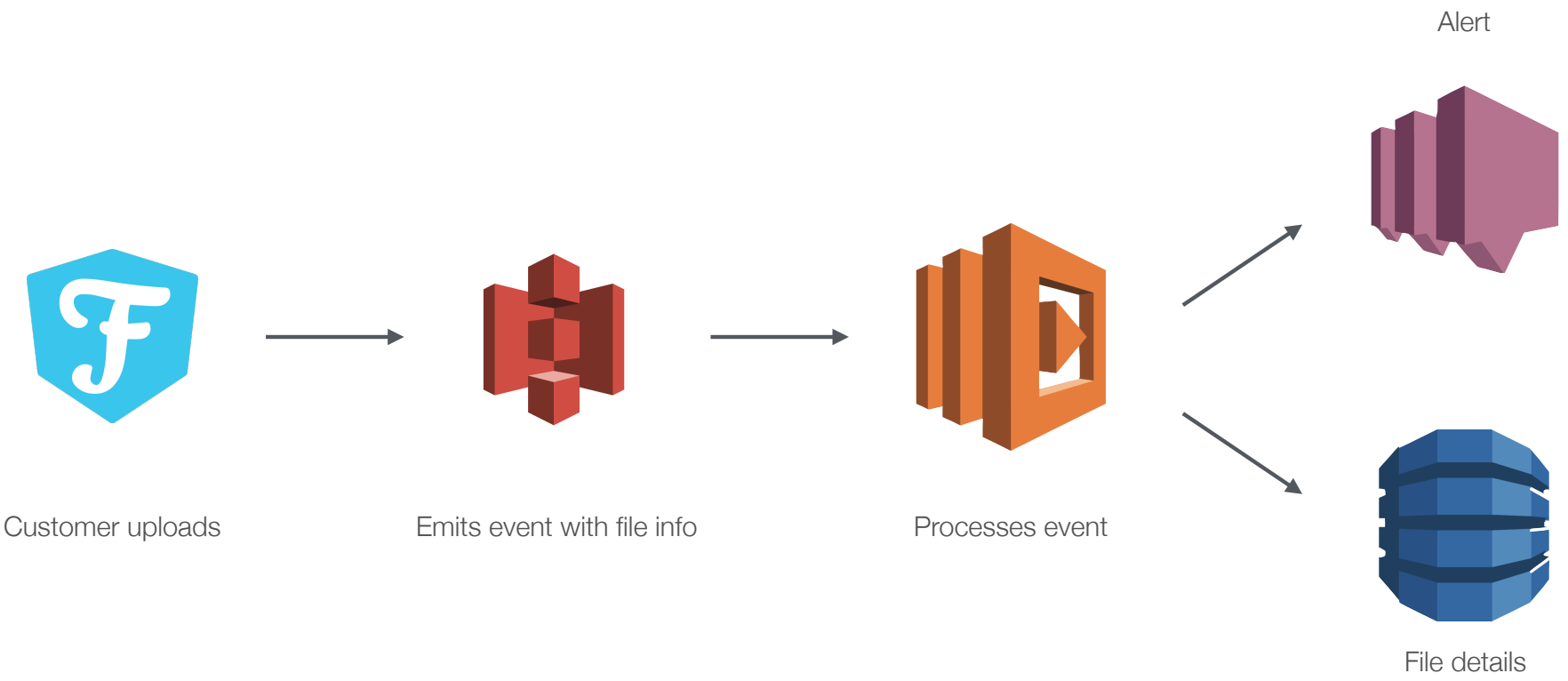
Logging and Monitoring Third–Party Services

Lambda

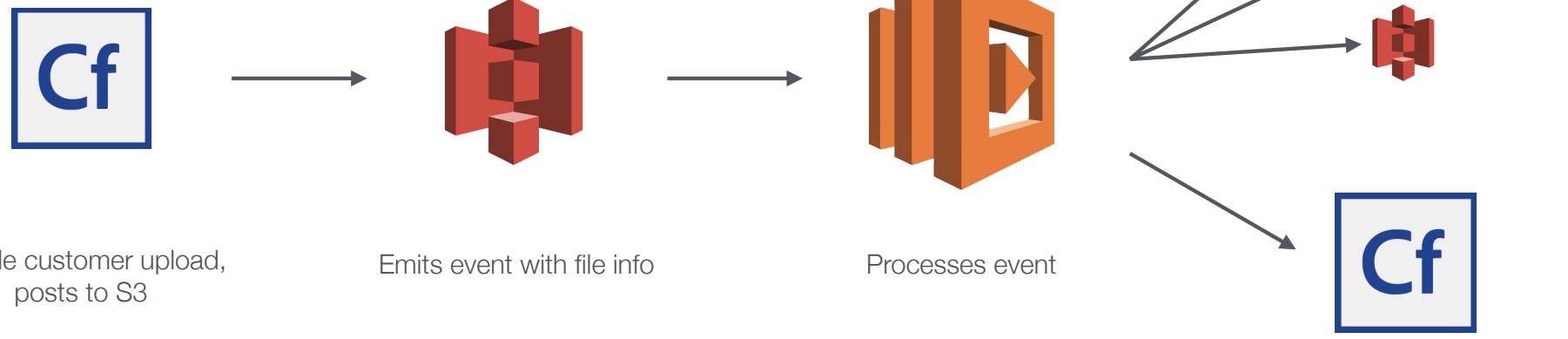
Logging and Monitoring Third–Party Services

Lambda

Logging and Monitoring Third–Party Services



Async Image Resizing



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

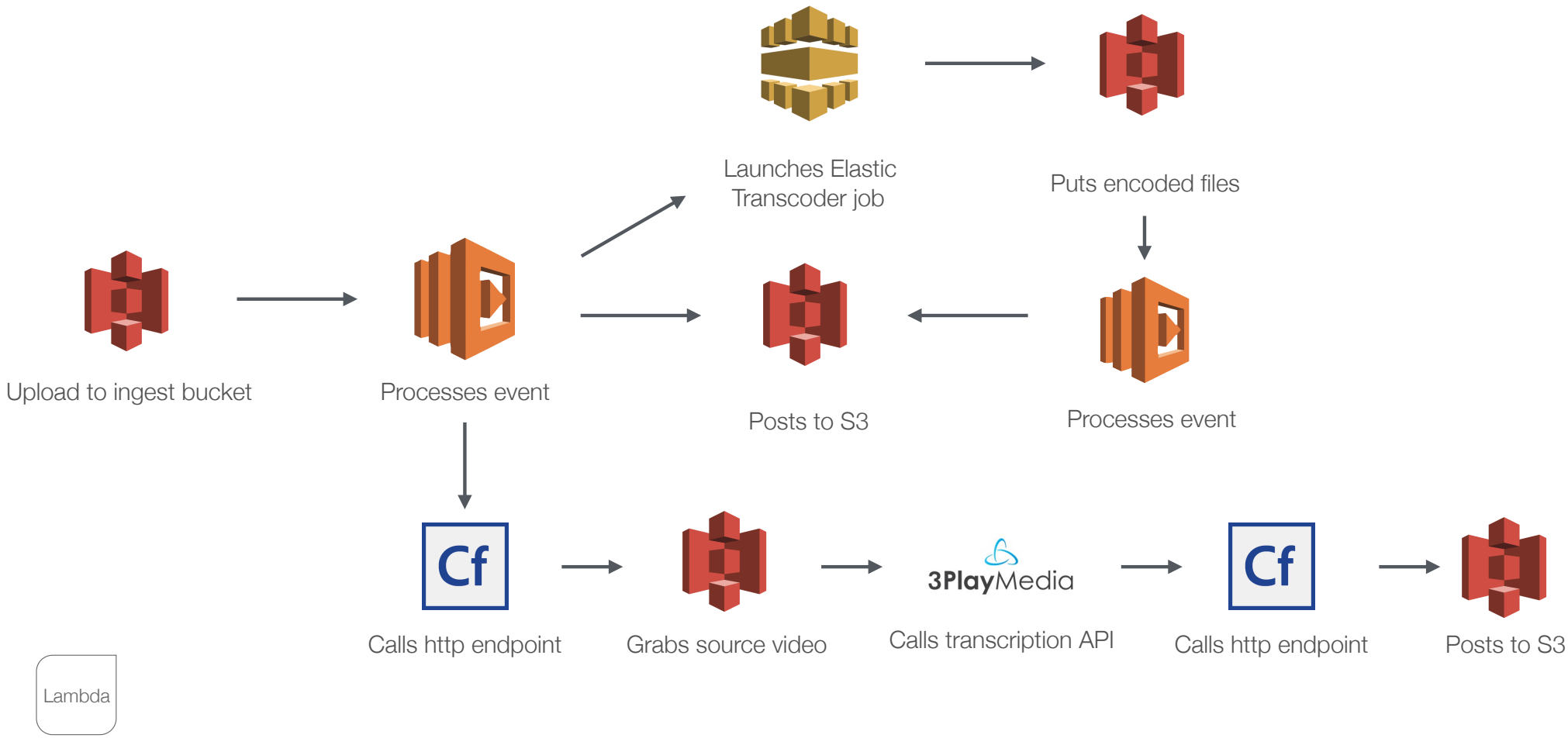
Puts encoded files

Processes event

Posts to S3

Lambda

Video Production Workflow (v2)



Demo

Invoking Lambda functions from 

github.com/brianklaas

Get the latest AWS Java SDK working in CF10+

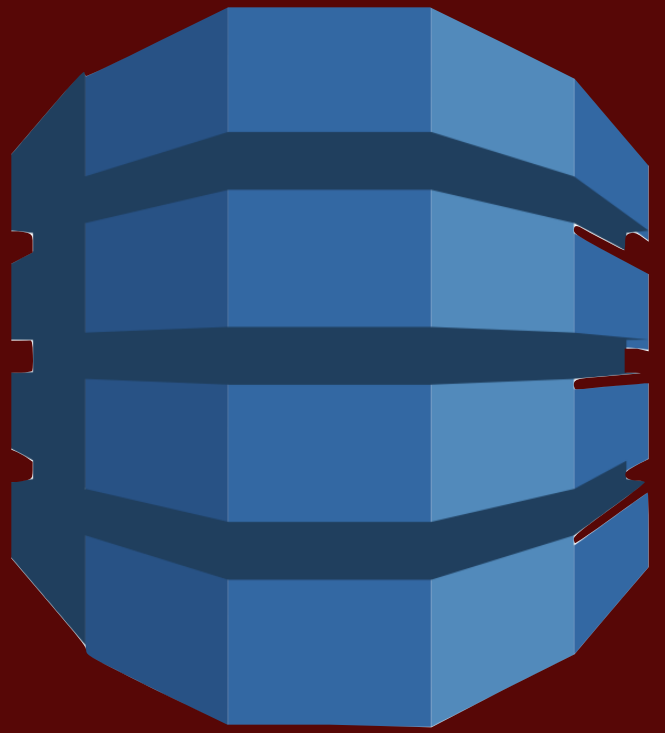
Add to your cfinstall/lib directory:

aws-java-sdk-x.x.xx.jar
jackson-annotations
jackson-core
jackson-databind
joda-time

= Focus on building apps,
not infrastructure

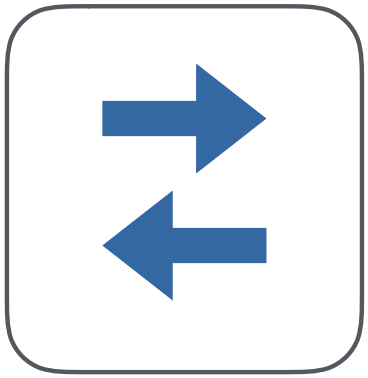
Lambda

DynamoDB



= Hugely scalable,
high-write throughput
document data store

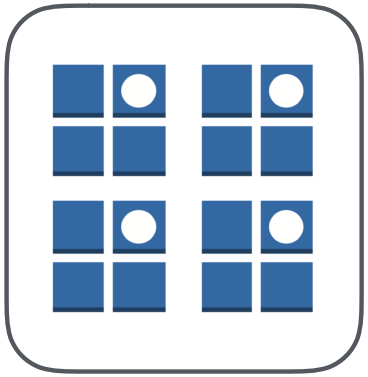
Dynamo
DB



1. Set read/write capacity

2. Set primary and sort keys

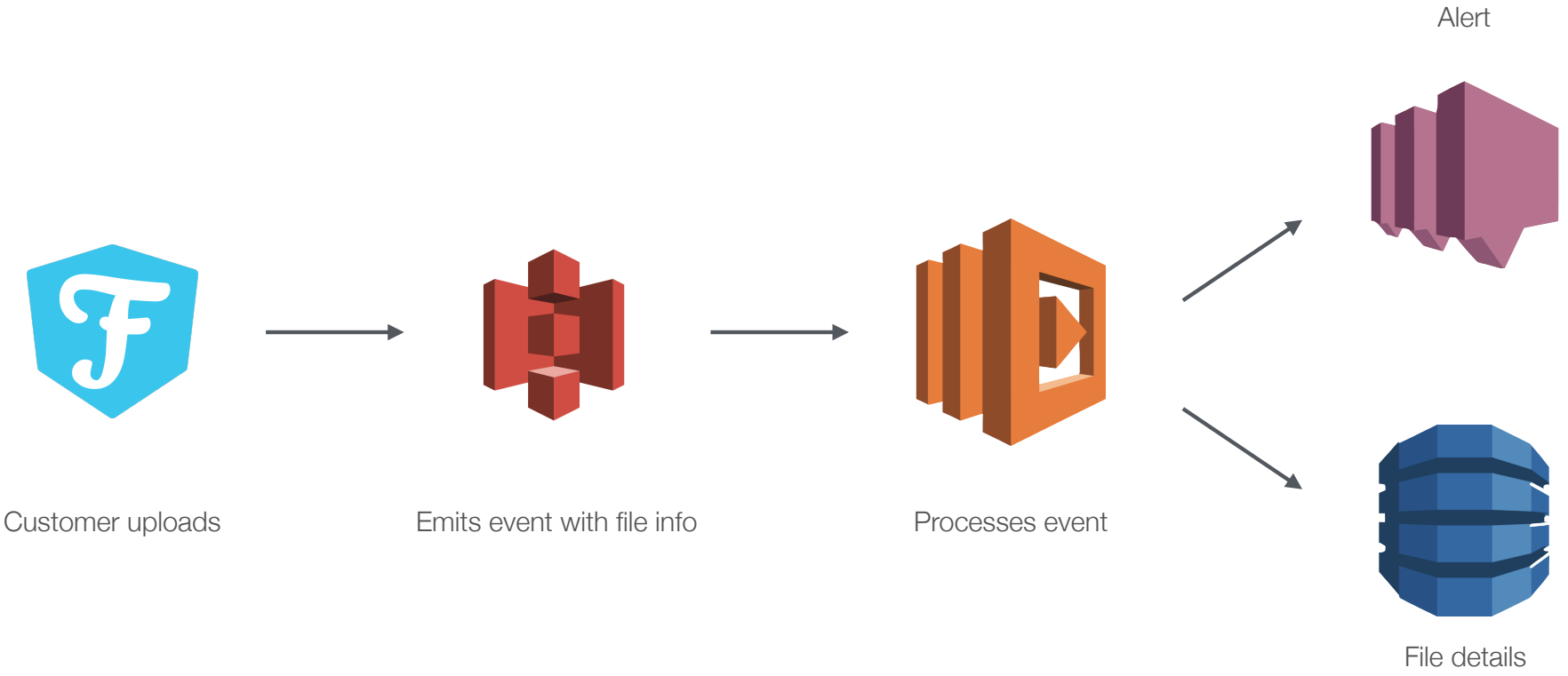
3. Set secondary indexes



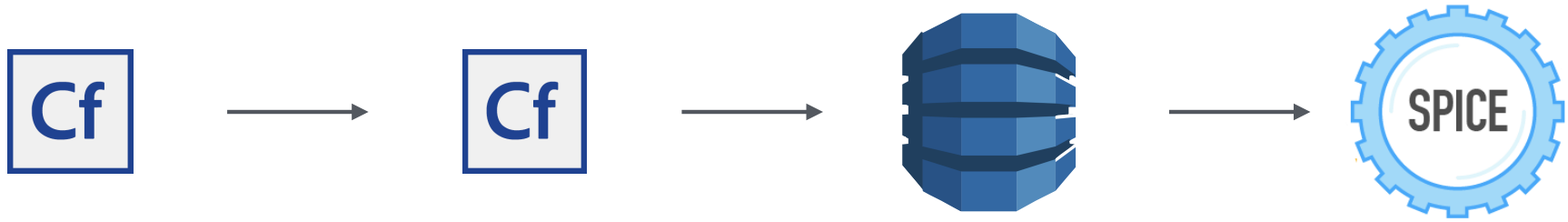
4. Write

Dynamo DB

Logging and Monitoring Third–Party Services



Exam Activity Analytics



Capture learner action

Writes batch data

Stores batch data

QuickSight

Dynamo DB

Demo

List, Put, Scan, Filter from 

github.com/brianklaas

Dynamo DB

Preparing for Outages

The Great DynamoDB Outage of 2015

Outages

Plan for outages.

(Blame Amazon)

Batch upload from your servers

Use multiple regions

Shut off services

Have a plan.


Go Do!

bklaas@jhu.edu

@brian_klaas

github.com/brianklaas