File Uploads

in Three Parts

@dgoodlad

Ops Lead at Envato

Part I – Infrastructure

You don’t want to do it.

Seriously.

capacity planning, 3am sms alerts, slow disks, full disks, slow network, flaky network, shared filesystems, backups, race conditions, ftp garbage collection, transactions, firewall bottlenecks, cdn, deprecated nginx file upload plugins, bandwidth costs,

Part II – Soft, Fluffy Clouds

Client-to-S3

... or Rackspace CloudFiles, or... or... or...

CORS; XMLHttpRequest2; File API

Better User Experience

Easy.

gem install awsraw

– or –

gem install aws

Your server authorises a client's request to upload

Client sends:

```
{
  “filename”: “not-porn.jpg”,
  “content-md5”: “d3adb33f...”,
  “content-type”: “image/jpg”,
  “content-length”: 31337
}
```

Server responds:

```
{
  “objectKey”: “temp/a-d3adb33f”,
  “objectUrl”: “http://s3.am...”,
  “headers”: {
      “Authorization”: “AWS ...”,
      “x-amz-date”: Time.now.httpdate
      “Content-Type”: ...,
      “Content-MD5”: ...
  }
```

Client performs a PUT

onComplete – Tell *your* server

S3 Lifecycle Management

Part III – What Now?

Media Processing?

Validation?

File Analysis?

Sure. Same as always.

Delayed Job, or Resque, or Sidekiq, or or or...

Part [OffByOneError]

Working Example

http://s3-photo-upload-demo.throwawayapp.com/
https://github.com/dgoodlad/s3-photo-upload-demo

Say Cheese!

Questions?