

A black and white photograph of a nuclear explosion's mushroom cloud. The cloud is massive and billowing, with a dark, dense column rising from the ground and a large, white, cauliflower-like top. The background is a dark sky, and the foreground shows some lower-level clouds or smoke.

Dr Clonelove:

HOW I LEARNED TO STOP WORRYING AND LOVE THE CLOUD

Using Amazon's Elastic Computing Cloud (and a few too many Star Wars analogies) to create a scalable server infrastructure for Drupal.

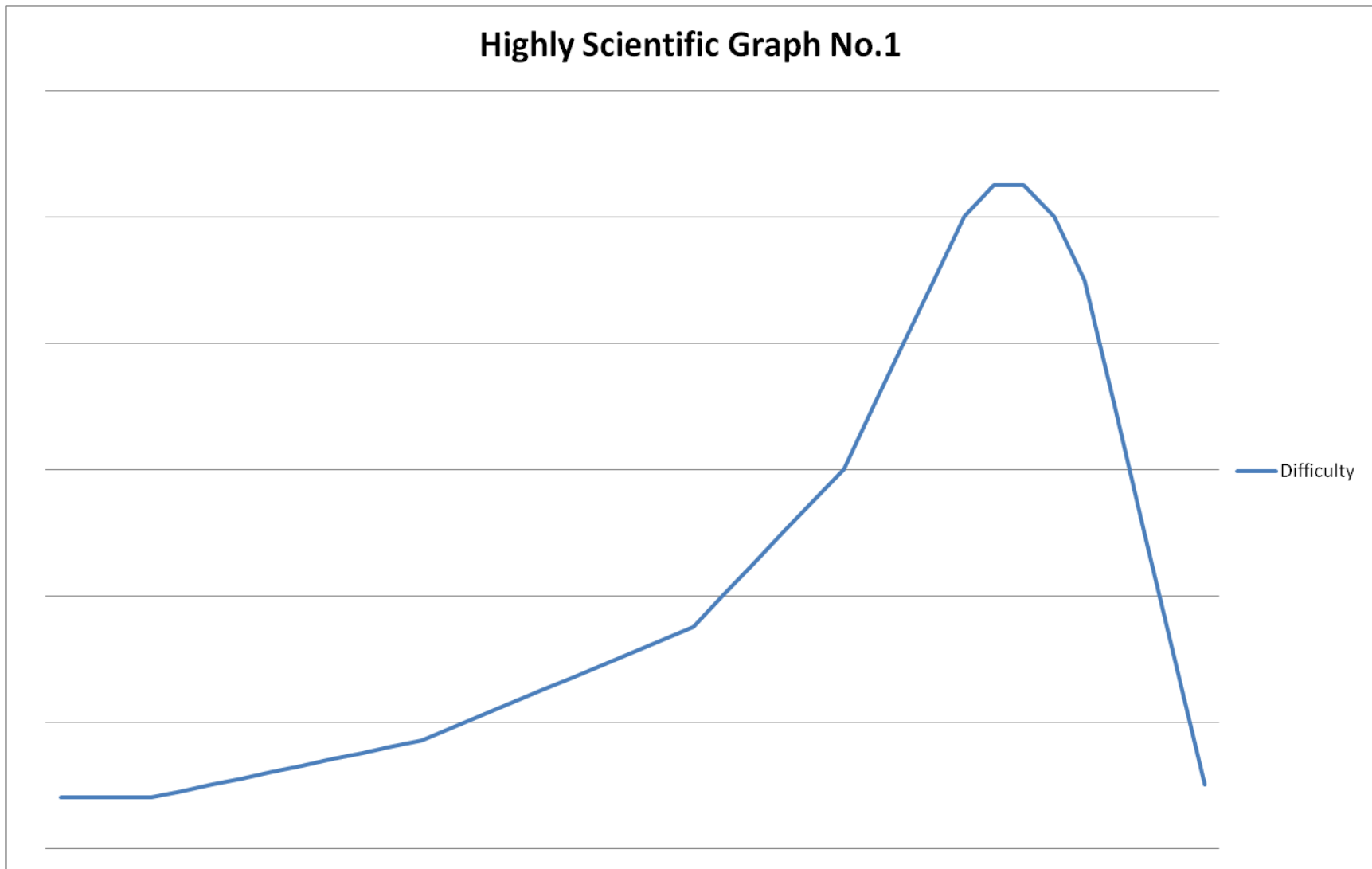
David O'Dwyer (T4partnership)

Peter Brownell (Code Positive / School of Everything)



Will help build
battle station
for food 🌌

Highly Scientific Graph No.1

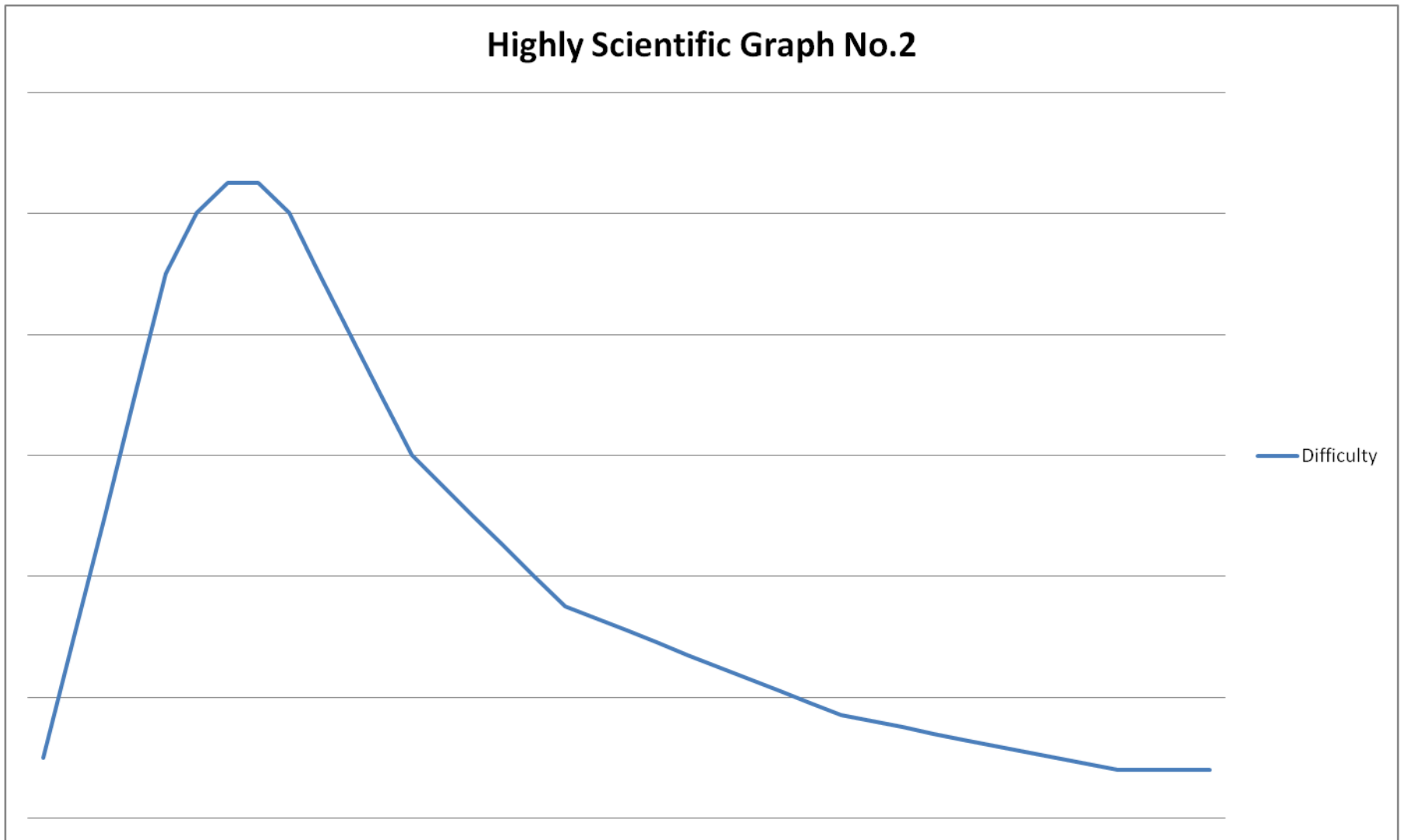


GETTING BIGGER

The normal scaling graph: starts easy, but gets harder over time

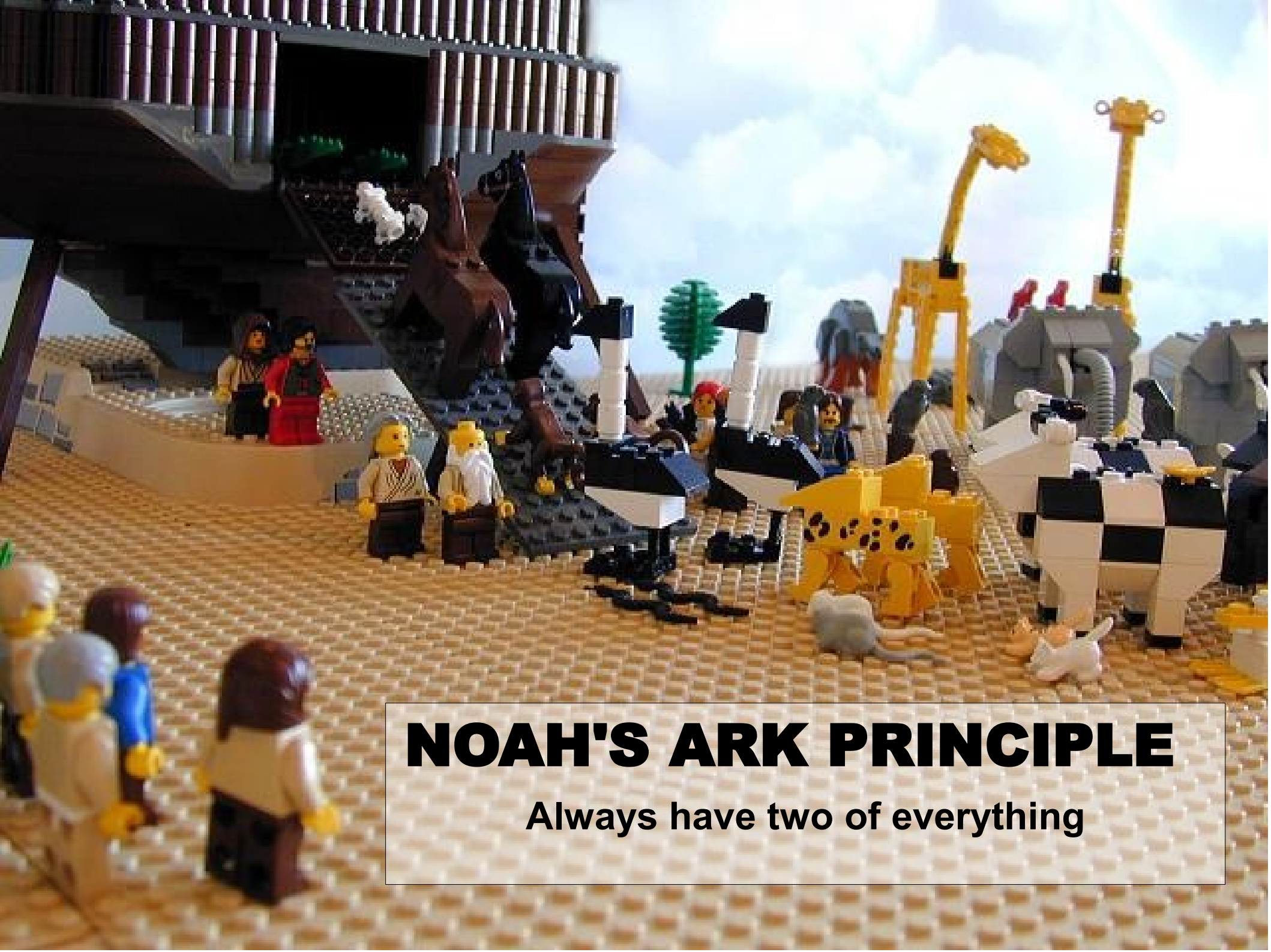
Scaling is not just a technology issue – your team and methodology must scale too

Highly Scientific Graph No.2



SCALING IS MORE THAN TECH

Drupal and EC2 force you to deal with details at the beginning
How do extend someone else's code, how to do you cope with a crash

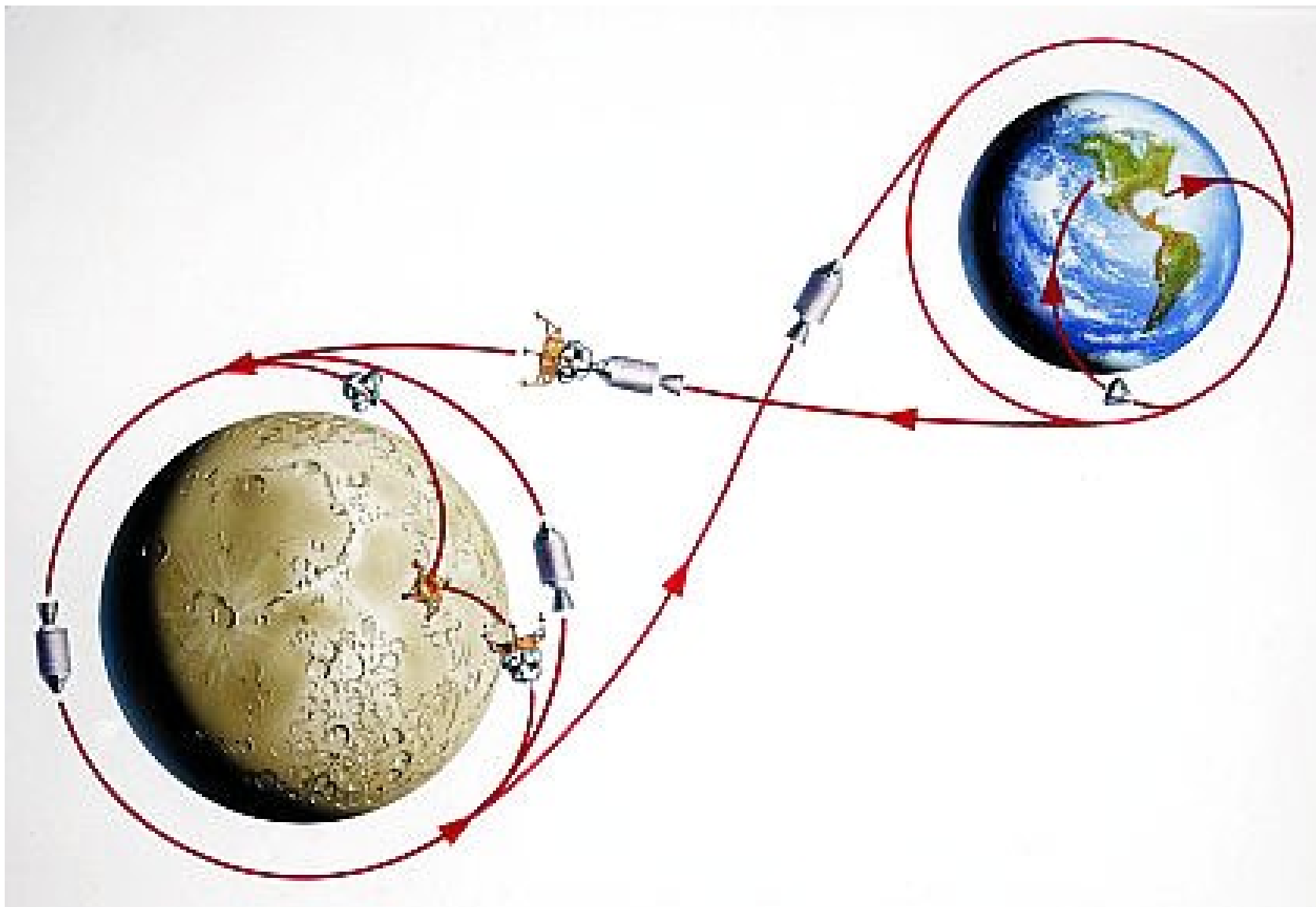


NOAH'S ARK PRINCIPLE

Always have two of everything

IT'S MUCH EASIER WITH CLONES





LAUNCH, ORBIT, RECOVER

Get it up there, keep it up there, get it down safely and know where it is all the time



CLOUD CITY

Have a base outside of the cloud

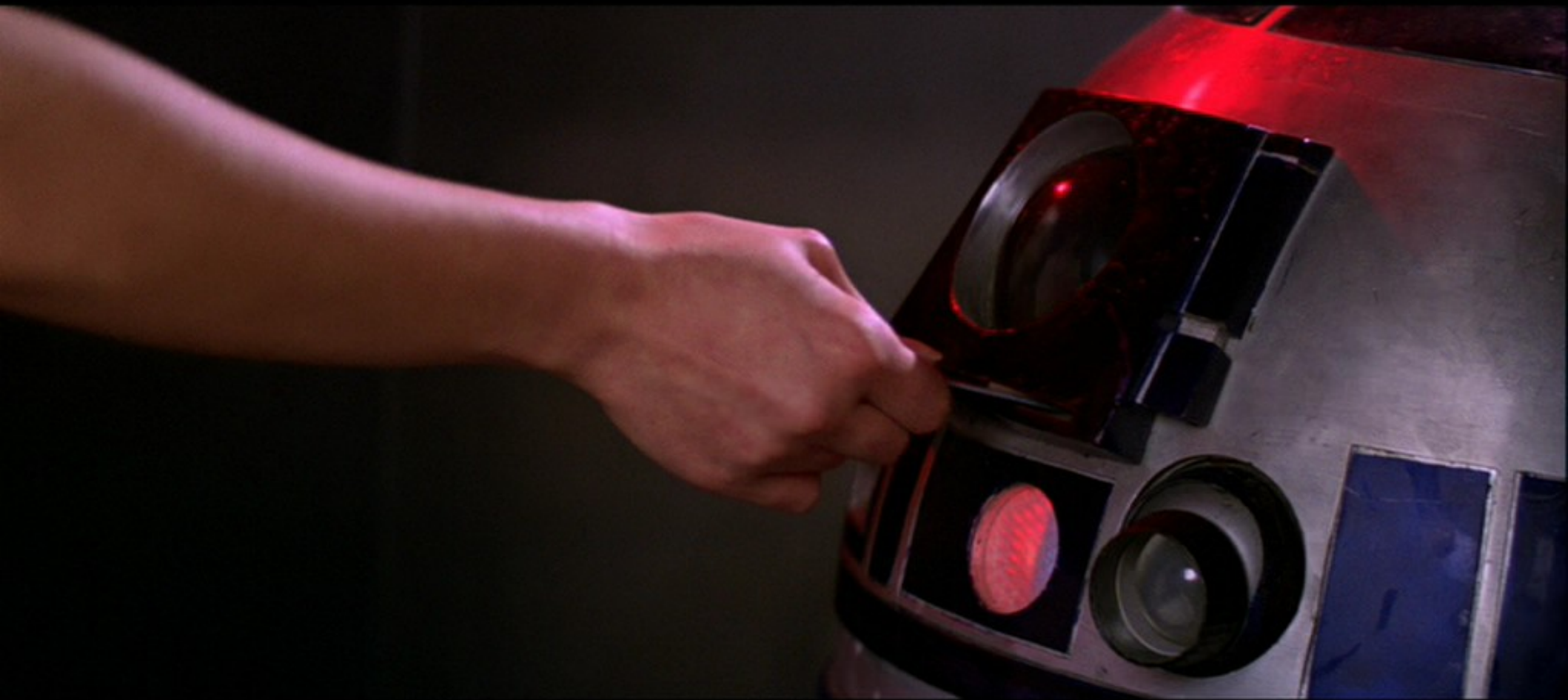




KNOW WHAT IS GOING ON

What do things look like when they are working

What do they look like when they are wrong



STORAGE YOU CAN RELY ON

EC2 currently only gives you disk space while your machine is running. When you turn a machine off, your data is gone.

We use ElasticDrive to mount a disk directly off S3.



C2C (clone to clone) FILESYSTEM

Avoid any central dependencies, and extra server costs.

Each server maintains a local copy of uploaded files.

GlusterFS runs on each machine sharing files as they arrive.

Primary webserver syncs files to S3



WHEN YOU CAN'T GET BATMAN

Go with Robin. Round Robin DNS is the cheap solution for load balancing. When load gets high, we can create a new load balancer clone.



Server management
The recommended server management tool is ElasticFox, a Firefox plugin to make it easier to keep track of images and running instances.

Webserver initialisation

(Parameters specified at boot can be obtained via <http://109.254.169.254/v1.0/user-data>)

cloud-init:specify-primary
cloud-init:specify-primary
cloud-init:specify-primary:file:schoolofeverything.com

On startup, a webserver will download its machine config from the repository, run export --force sam-sch/config

Then, any package updates are made, via APT. Critical package versions are "pinned" in the config. After installing a new package run sudo dpkg-query -f='\${Package} \${Version} \${Architecture}\n'

Next we override any default configs by checking out a machine specific application config from the repository.

The website is then extracted from version control and Apache restarted to make it active.

Files are automatically synchronised with S3 on the primary server, and via GlusterFS on other websevers.

New routes will need to be added to the glusterfs server config.

Server should now added to the Round Robin dns pool. This should be done manually once the server is operational.

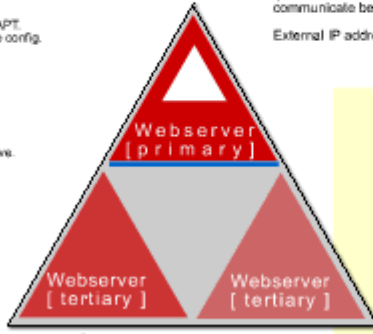
Machines will need to be updated to get their external IP config (good plan anyway)

External Email

The primary webserver is also primary-1.mail@schoolofeverything.com it will send all the outgoing email. Externally, this server is seen as smtp.schoolofeverything.com

We secure our external mailbox. Each mail relay to smtp.p2555.tumblr.com through the mailhost (smtp-relay-b1.s1.amazonaws.com) We do this using Postfix to share our internal network with anything we send. So, this sends our mail, any mail we could receive through it, to you.

MySQL EC2 Instance Profile
All EC2 instances are registered with an EC2 Instance Profile to be able to connect to the database. <http://www.schloofeverything.com>



Round Robin DNS

New websevers can be added by editing DynDNS records. We will upgrade to a managed load balancing system when needed.

Round Robin DNS is managed via the advanced interface on DynDNS (found at www.dyn.com)

Round Robin records are A records, and they can be added or upgraded as necessary.

We have an IP address range for our managed IP addresses and a file that we update each server with a unique IP.

Dynamic DNS

All machines register themselves with dynDNS with ddclient. This will register their INTERNAL IP address. Our machines will use these to communicate between themselves.

External IP addresses are manually managed.



GlusterFS cluster file system
GlusterFS works in a master-slave mode by setting up each node as both a server and a client. Each node is a server for its own machine and a client for the other machines. Each of the nodes is a client for the other nodes on the network. It is possible to have a single node as both a server and a client.

Each webserver runs both a server and a client. The client connects to 127.0.0.1 and the servers manage the data on the client.

When a new server is added, the patterns config needs to be updated to make the servers aware of each other.

Elastic Drive Software RAID
Elastic Drive is a software RAID solution for EC2. It is a software RAID solution that runs on the host. It is a software RAID solution that runs on the host. It is a software RAID solution that runs on the host.

When a new machine is booted, we add the elastic drive 'fs' and the raid 'md'.

Relabelling Elastic Drive
When a primary server comes on, so for the first time, we need to relabel the elastic drive. This is done by running the following commands:

```
sudo mdadm --create --verbose /dev/md0 --raid-level=1 --raid-disks=2 /dev/sd*
sudo mdadm --grow /dev/md0 --raid-level=1 --raid-disks=2 /dev/sd*
sudo mdadm --detail /dev/md0
```

Linking RAID
So that our RAID system is usable as a single file system, we need to link the RAID system to the file system. This is done by running the following commands:

```
sudo mkfs.ext4 /dev/md0
sudo mount /dev/md0 /mnt
```

Weberserver Files Directory
Weberservers share the contents of their files directory via GlusterFS. This is a virtual filesystem that will monitor a directory, and synchronise any changes across members of the cluster.

Elastic Drive Files Directory
The primary server has its files directory mounted via Elastic Drive. This will automatically write any new files to S3 so that a new primary webserver will immediately have all files needed.



Controller



Image building



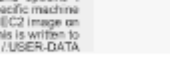
Aggregated Logging



Backup System



Website SVN Repository



Configuration SVN Repository



Daily off-site backups

Monitoring and Alerts
The controller machine will monitor all elements of the cluster. It will send alerts of any failure via SMS (Twitter).

Image building
Our server configurations are managed via XEN images on our controller machine.

Aggregated Logging
Any cluster system needs to have central logging. We use syslog-ng to send all our logs over the network. We can then examine all logs in one place, as if they came from the same server.

Website SVN Repository
Our live website is stored in version control. Any code change or modification can be done on a development or testing machine and then checked in. We extract updates on to the websevers from this repository.

Configuration SVN Repository
Weberservers share the contents of their files directory via GlusterFS. This is a virtual filesystem that will monitor a directory, and synchronise any changes across members of the cluster.

Daily off-site backups
Once a day we pull our backups off amazon completely and store them on a completely separate system. This server could be brought up as an emergency version of the site.

Database Naming

A databases with the following names:
primary-1.database.schoolofeverything.com

The first primary database is on the primary-1 database. This is the primary database.

The first secondary database is on the secondary-1 database. This is the secondary database.

The first tertiary database is on the tertiary-1 database. This is the tertiary database.

Check for corrupt tables with `mysqlcheck --user='mysql' --host='127.0.0.1'`

Database Failure

In the event of a primary database failure, we have all data on S3.

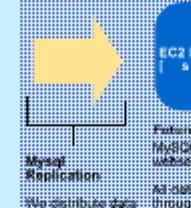
In the event of a primary database failure, we have all data on S3.

Starting a new primary image will reconnect to the S3 store, and be back in less than 5 min.

Check for corrupt tables with `mysqlcheck --user='mysql' --host='127.0.0.1'`



MySQL [primary]



EC2 Database [slave]

Future Database Plans: MySQL Proxy
MySQL proxy will run on each webserver.
All database queries are passed through a proxy. This will allow us to use Read Replicas (MySQL 5.6) and be ready to scale servers if necessary.

Write Backups

The Elastic Drive on the primary server will write to S3. This means that we have a backup of our data on S3.

We use a virtual filesystem to keep a live backup of the database on S3. This means that we have a maximum data loss of a few seconds if something should happen to the database server.

Differential Backups

We create a differential backup of the system every hour. This is done by using the system's backup utility. This means that we have a backup of our data on S3.

In a system crash, a backup of the system is available. This means that we have a backup of our data on S3.

All items on S3 have a url. We perform our backups with aws tools.

S3 Live Storage

S3 Downloads

All items on S3 have a url. We perform our backups with aws tools.

