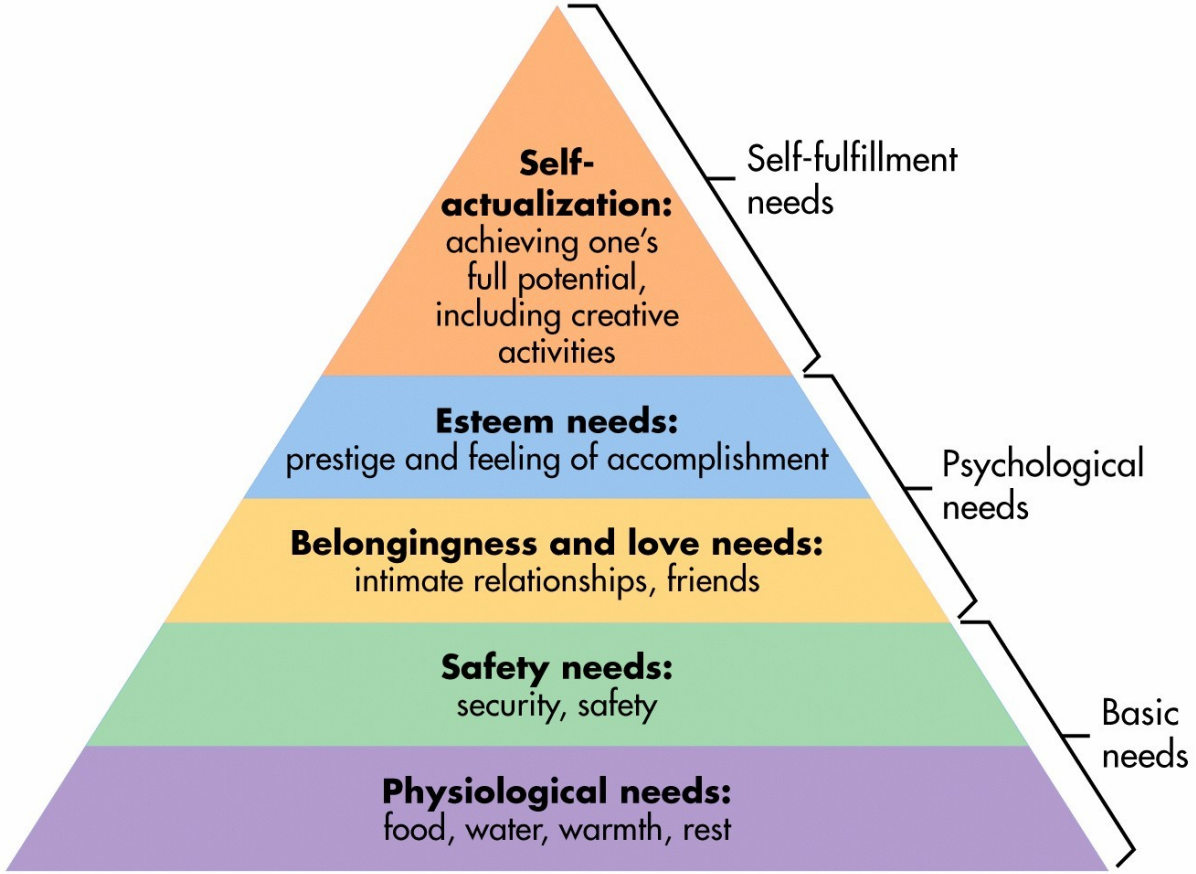
Dropbox

It will break!

Leonid Movsesyan, Dropbox

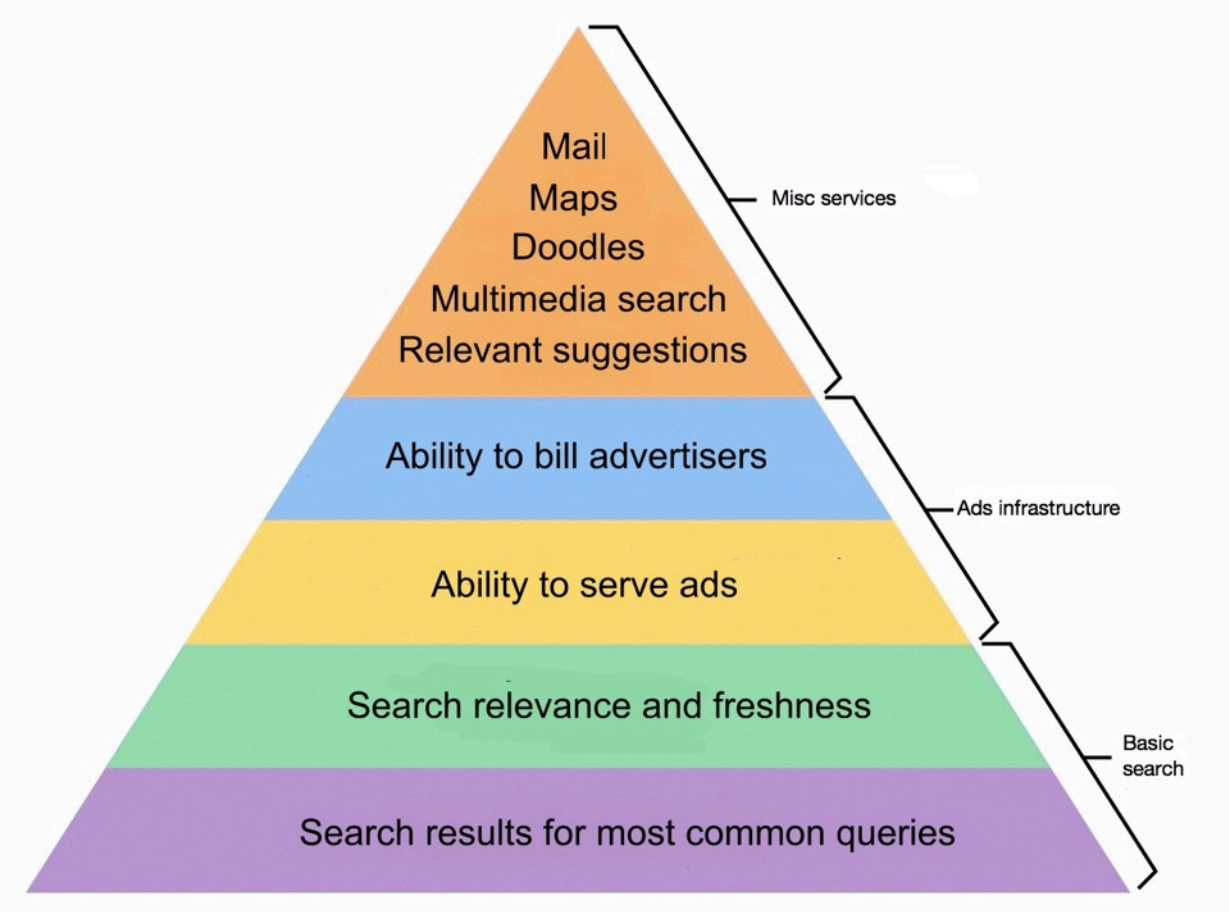
Hierarchy of needs



4 easy steps

- Build your hierarchy of needs
- Put together your assumptions about the each layer
- Run regular disaster recovery testing (DRT) as close to reality as possible
- Adjust them constantly to reflect the changes

Search engine example



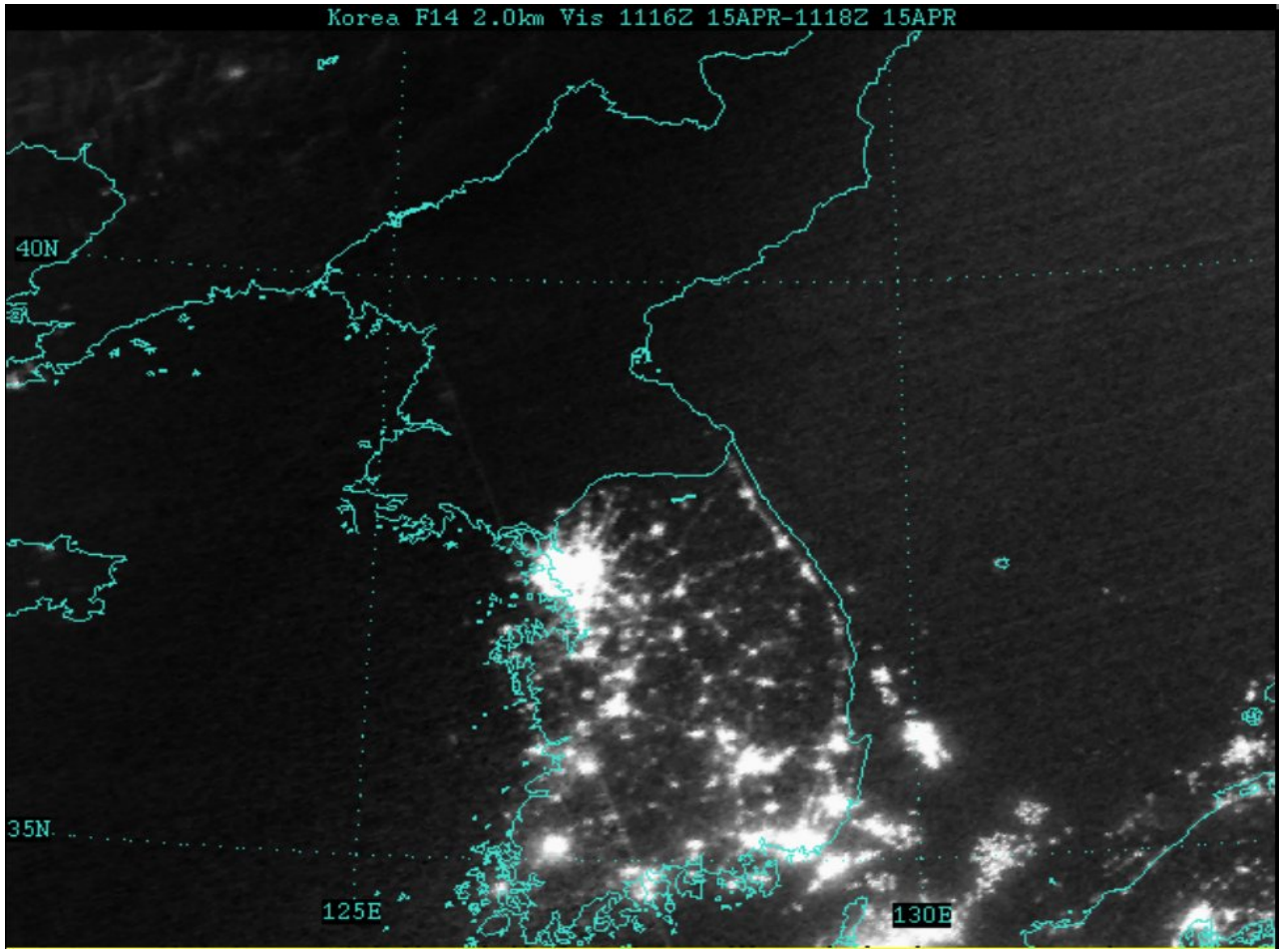
Optimize for metrics that you care about

Treat your DRTs how you treat your unit tests

How to design a good DRT?

- For every action in your design ask yourself ‘What if?’
  - We’ll use S3 to store our data: what if S3 availability zone go down?
  - We’ll run credit card processing using this 3rd party vendor: what if it times out?
  - We’ll store the metadata in MySQL: what if MySQL master will die?
- Go with this question as deep as possible and use ’what if?’ scenarios as a DRT

Power outage



Power will fail

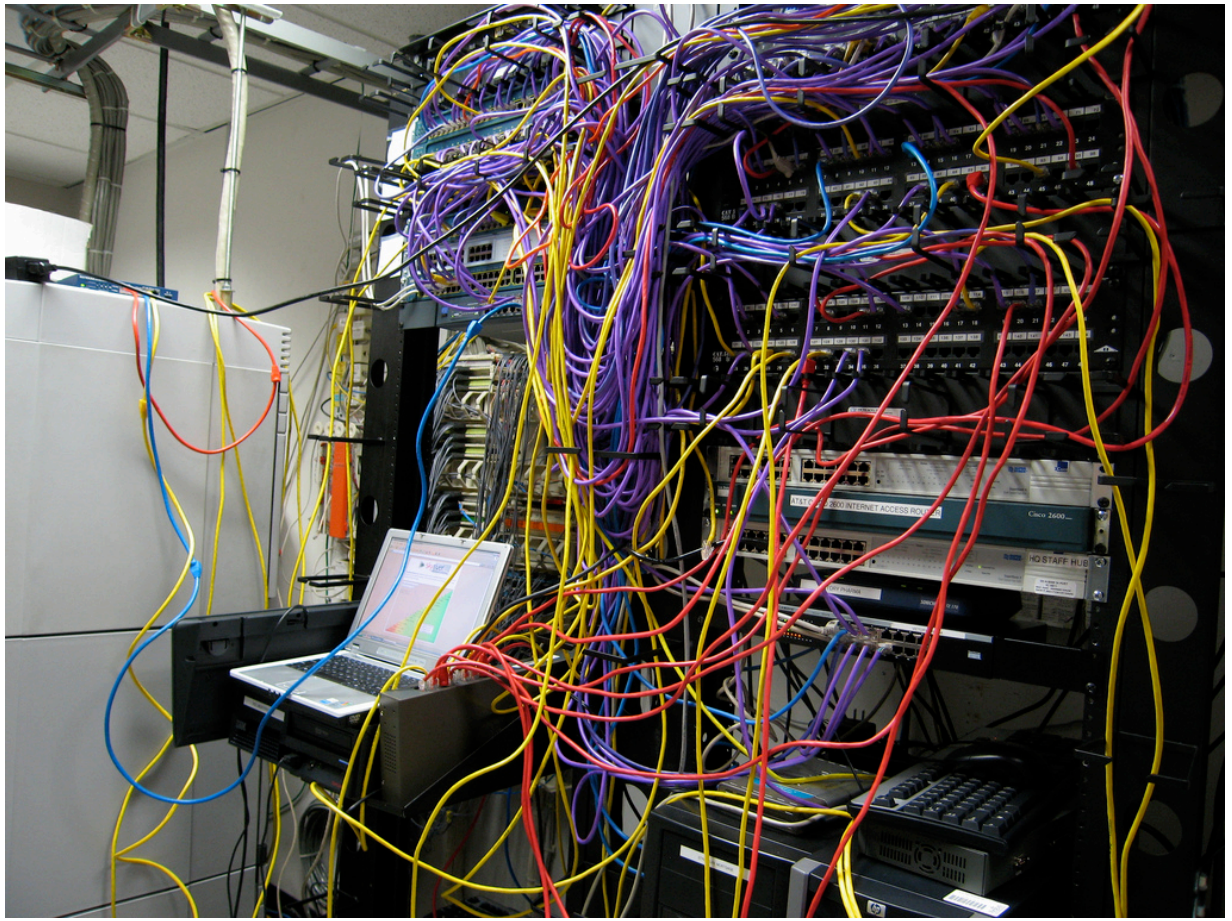
Diesel generators may not start

You’ll run out of diesel sooner than you expect

Avoid the consequences

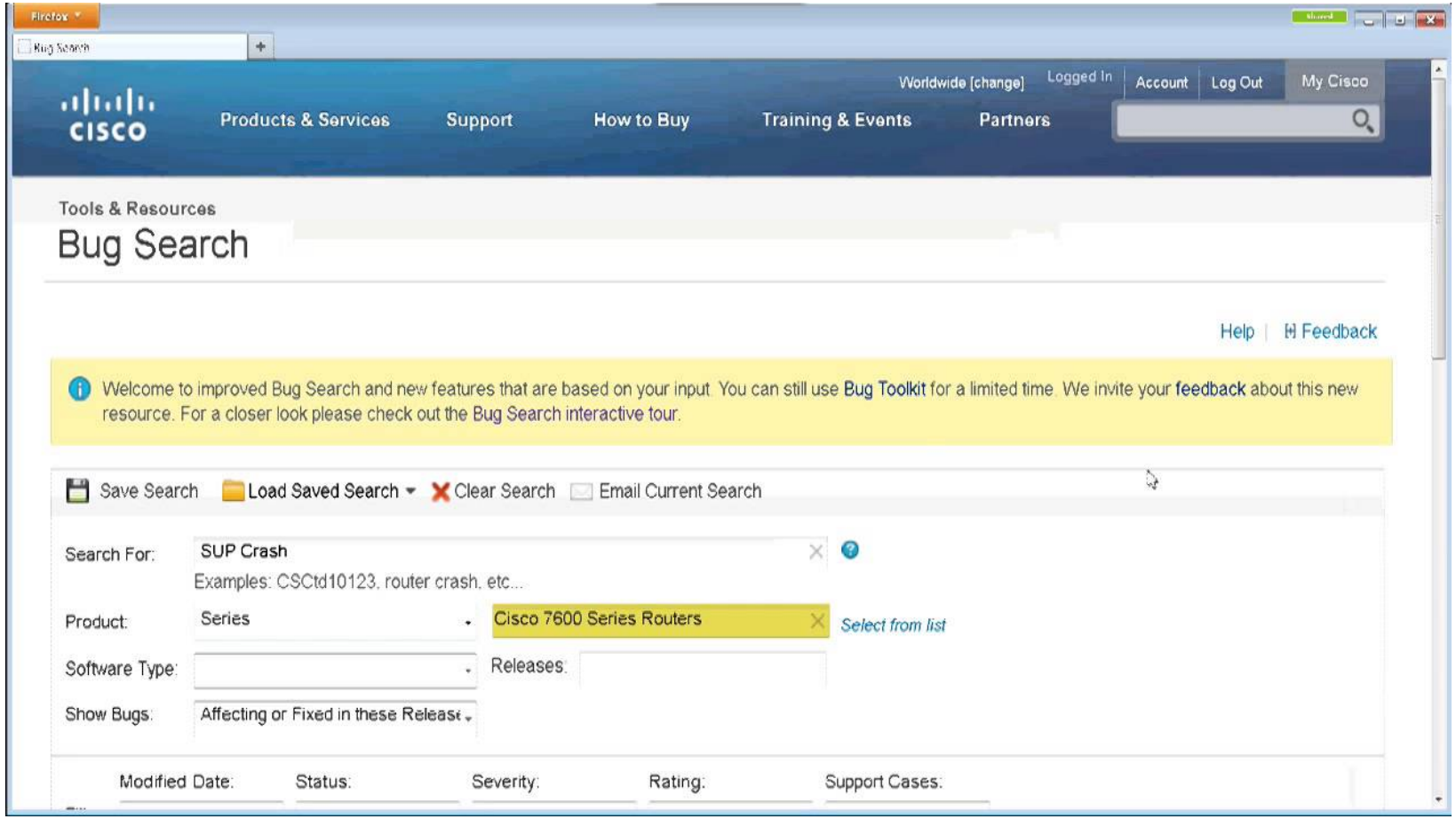
- Split the servers by groups based on the hierarchy of needs
- Create automation that will allow to power off the top of the hierarchy first
- Test this automation regularly as well as your diesel generators

Network outage

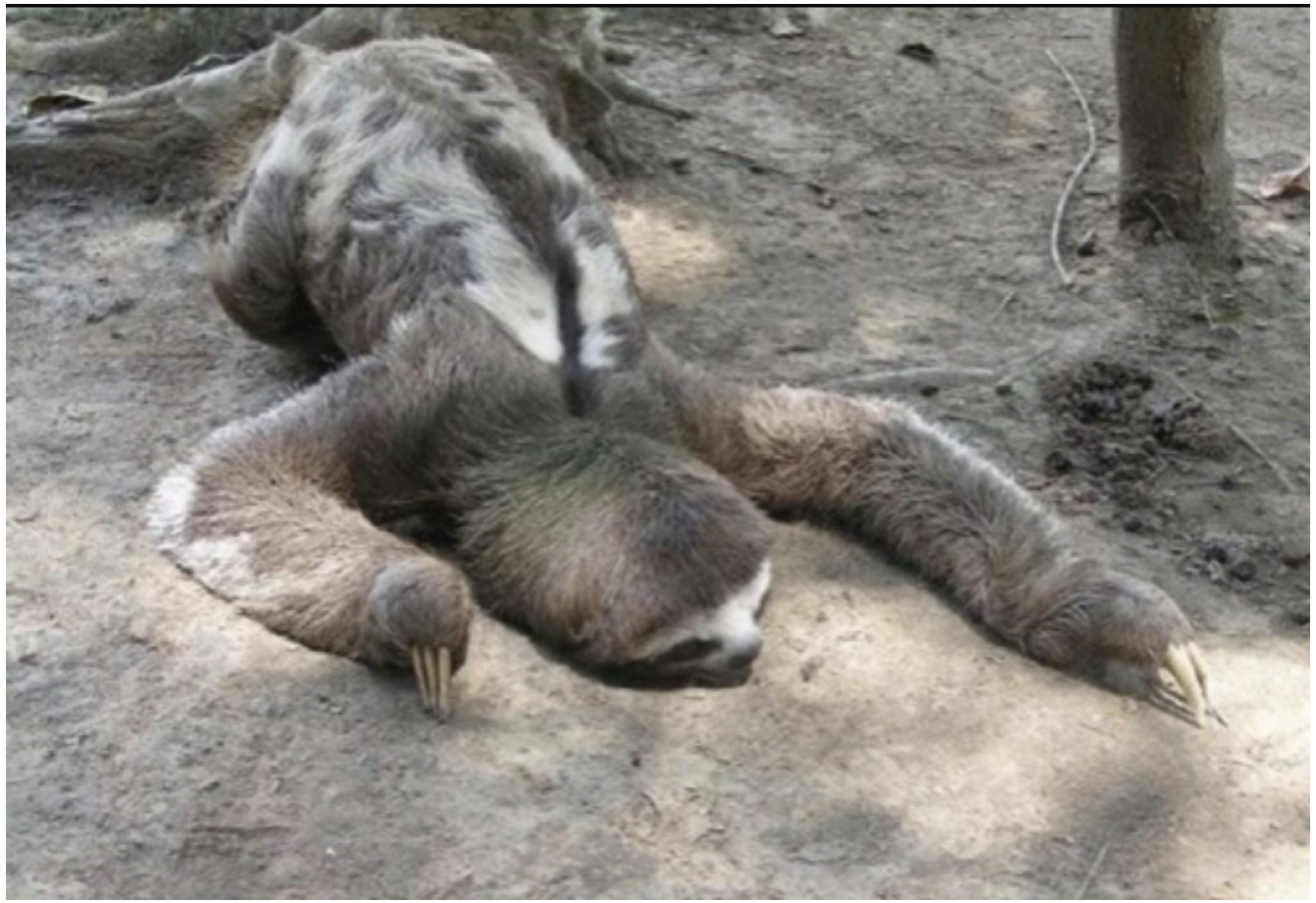


Never expect networks you don’t control to be reliable

Never expect you switches on all the levels of your network to be always available



Expect the network to fail slowly



Network testing

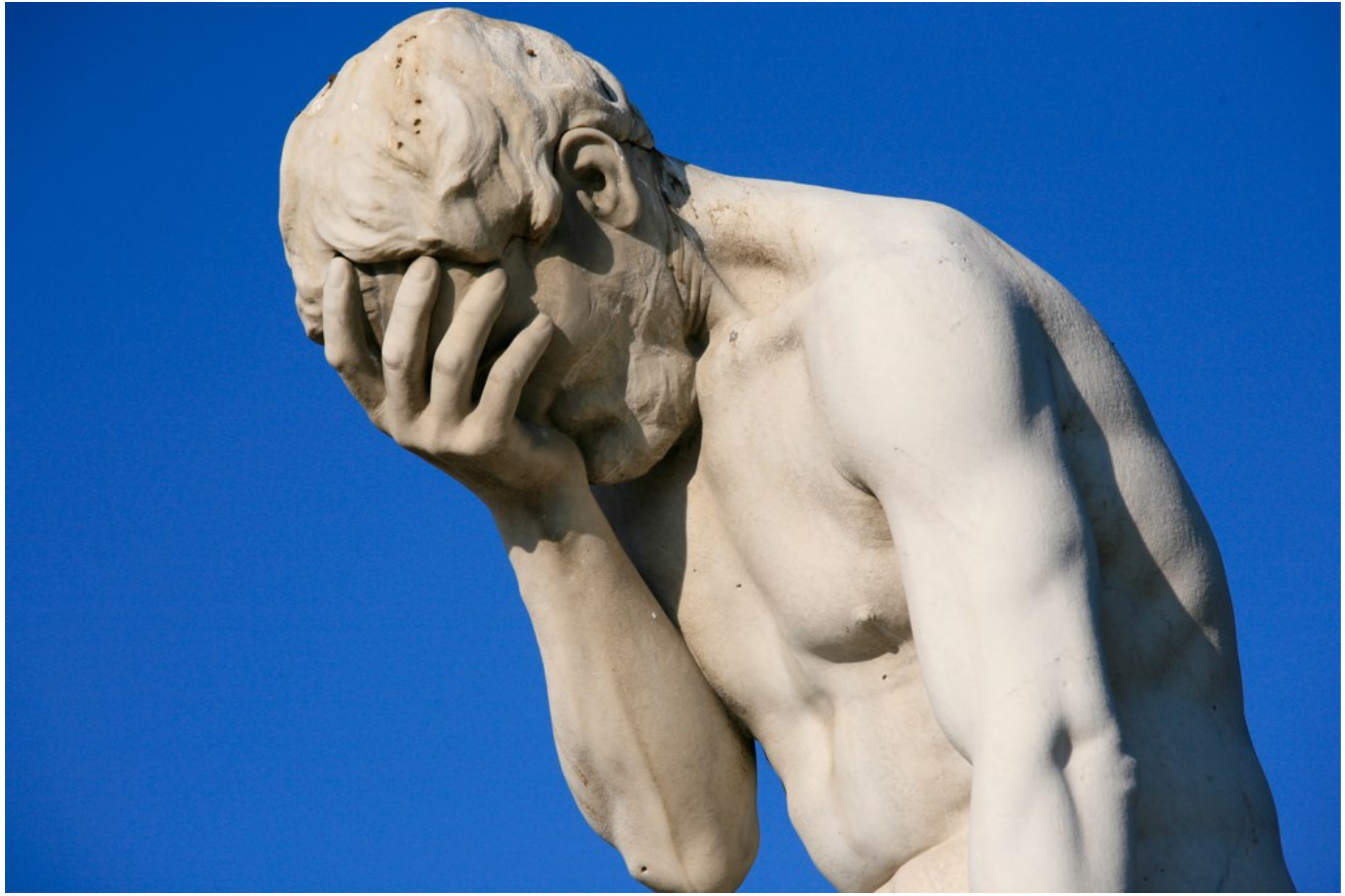
- Use DRTs to fine tune timeouts
- Imitate multiple types of network issues
- Failover every network device in your own topology

Pro tip: Use switch failover as an opportunity to upgrade the firmware

Wishful thinking

```
for log_file in log_files:
        send_log_file_to_s3(log_file)

for log_file in log_files:
        gzip_log_file(log_file)
        delete_log_file(log_file)
```



Cloud vendors

- Expect your cloud instance to fail
- Expect you cloud stored data to get lost or corrupted
- Expect to lose network connectivity to your cloud provider
- Expect cloud providers to lose the whole region

Try not to over-engineer

Don’t forget about human error

Avoid any manual operations and runbooks in places where the mistake can not be tolerable

Analyze and group you outages

- Structure the root causes of the outages
- Analyze times to detect, diagnose and recover
- Group outages to identify the patterns

Questions?