Trust Me, I’m a Cloud
Who Am I?

• Sam Macleod
• Security Consultant
• Former Ops Manager
• Disaster Survivor
Disclaimer

- My opinions are my own
- Not necessarily those of my employer
- Should not be considered a replacement for expert advice
Summary

• Business continuity problems that are solved
• Business continuity problems that are not solved
• Same old problems, brand new look
• Fault Tolerance vs Disaster Recovery
• When SaaS betrays us
• Vendor lock-in
• Integrity and Availability
What is The Cloud?

- SaaS products that we use in our organisations
- IaaS solutions that we build our applications on
  - Operating systems
- PaaS solutions that we build our applications on
  - Applications
- Serverless technologies that we build our applications on
  - Function
Not All Are Created Equal

- Some make excellent service level commitments (many 9s)
- Some meet or exceed their guarantees
- Some have no SLAs at all
- Do we have a process for choosing vendors?
Things That Are Better

- Fault tolerance
- Capacity management
- Scalability
Things That Are Not Necessarily Better

- Disaster Recovery
- Possession of Data
- Vendor Lock-in
What are the risks?

- Catastrophic loss of data
- Service outage
- Account lockout and takeover
- Service discontinued
How can these issues manifest themselves?
Software as a Service

• How much do we use?

• What happens if it goes down?
  • Can we deliver service?
  • Can we operate our business?

• Can we retrieve our data?

• Can we move between service providers?
Who possess our data?

- We may or may not retain legal ownership of it
- Can we backup or export our data?
- Can we move our data from a cloud to an on-premise solution?
- What are we storing?
- What are we consuming?
- What are the worst case scenarios?
What can go wrong?

• What services do our applications rely on?
  • What components rely on SaaS?
  • Can we deploy?
  • Can we monitor?
  • What security products do we use?
  • What might we lose in an outage?

• What services do our organisations rely on?
  • Can we do our work?
Infrastructure and Platform as a Service

• What failures can we tolerate?
• What failures can we survive?
• How fast can we recover?
• What are we not protected against?
Fault Tolerance

• How much of our application or environment can fail before service is impacted?
• How can we automatically recover from failures?
Disaster Scenarios

• Loss of an instance
• Loss of an Availability Zone
• Loss of a region
• Multi-region outage
• Loss of an account
• Loss of integrity
Design Solutions

• Multiple Availability Zone
• Multiple Regions
• Multiple Accounts
• Multiple Providers?
When Replication Isn’t Enough

• Replication is a core concept in Disaster Recovery
• Changes to a primary service are replicated to a stand-by service
• Can be built in, logical or physical
• Can also propagate failures throughout the environment
Serverless Architecture

• What serverless technology do we use?
  • Lambda
  • S3
  • API Gateway
  • SNS

• Could we operate our applications without these vendor specific technologies?

• Could we move to another provider if we had to?

• What would the cost be?

• How long would it take?
Back to Reality

• What are the costs?

• $$$

• Management overhead
  • How much does it cost to maintain?

• Complexity
  • How hard is it to make changes?
  • How much can be automated?
  • How reliable can we make it?
How do we prioritise?

- What services need to be available at all times?
- Are we regulated?
- What data is critical to the organisation?
- How long can we afford to be down?
- What could we afford to lose?
- What are our single points of failure?
Planning for survival

• What can we do in a worst case scenario?

• If we had to rebuild, how would we do it?
  • Recover from backup.
  • Redeploy
  • Can we deploy onto a different platform if we have to?
Objectives

• Recovery Point Objective?
  • How much data can we sacrifice?
  • How far back can we go?

• Recovery Time Objective?
  • How long will it take to do it?
When to act

• How long should you be down before activating DR plans?
• How do you evaluate the length of an outage?
Can we test it?

- Are our plans fully documented?
- How much is automated?
- Is it possible to perform tests?
- How often should we do it?
Recap

• Do we have visibility of all of the different cloud services we use?
• Do we understand which ones are the most critical?
• Do we know our single points of failure?
• How much do we rely on vendor guarantees?
• Do we understand our priorities, and do we agree on them?
• Have we designed our systems to sufficiently address these risks?
• Can we respond to a worst case scenario?
• Are we able to test these systems?