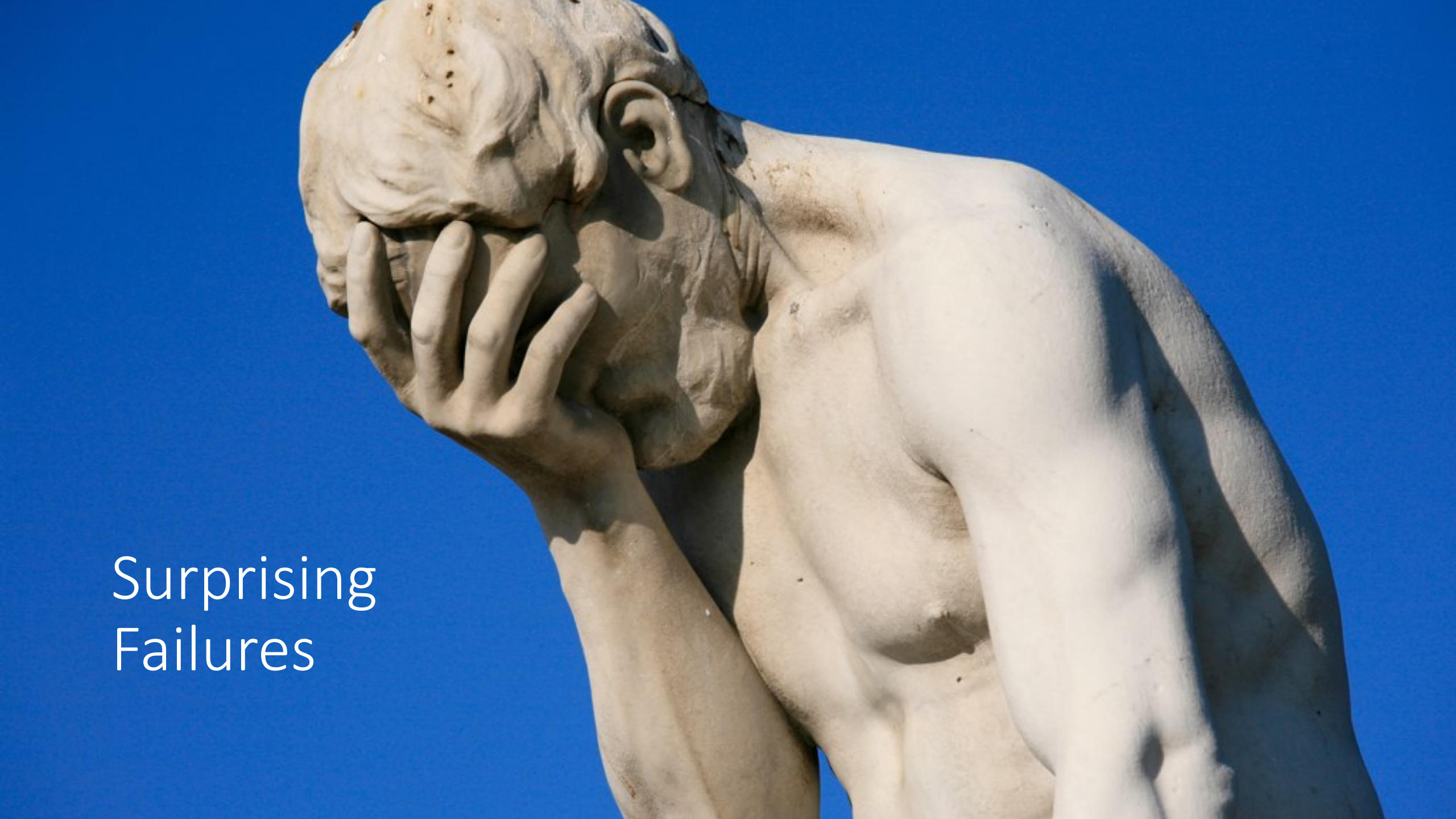
Modern Cloud Fundamentals

Christopher Bennage

patterns & practices
AzureCAT


Surprising
Failures

Scaling failures

- Unexpected behavior of libraries at scale
- Unnecessarily complicated networking topologies

Resiliency failures

- When a non-critical dependency failed, it brought down the main service (lack of circuit breaker)
- Running a critical workload on a single VM

some thoughts to consider

The physics has changed

- The architectures and methods that used to bring success don’t always work in the cloud.
- Even worse, they will *appear* to work until some critical event.


Performance
Testing

Fallacies of Cloud Computing

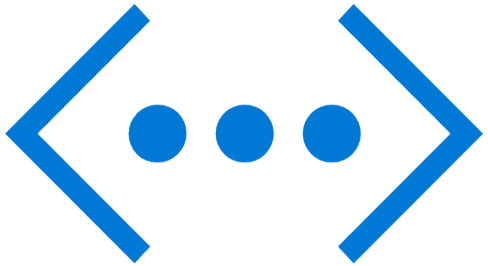
1. Everything is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. Security is inherited
5. Topology doesn't matter.

The Myths of Lift & Shift

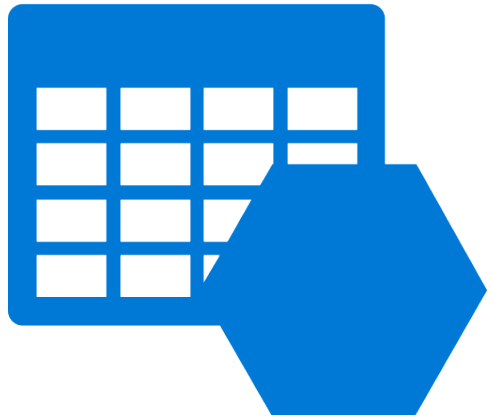
If you *just* migrate your workload to the cloud…

- It will perform better
- It will scale out easily
- It will be more reliable
- It will cost less

Compute



Network

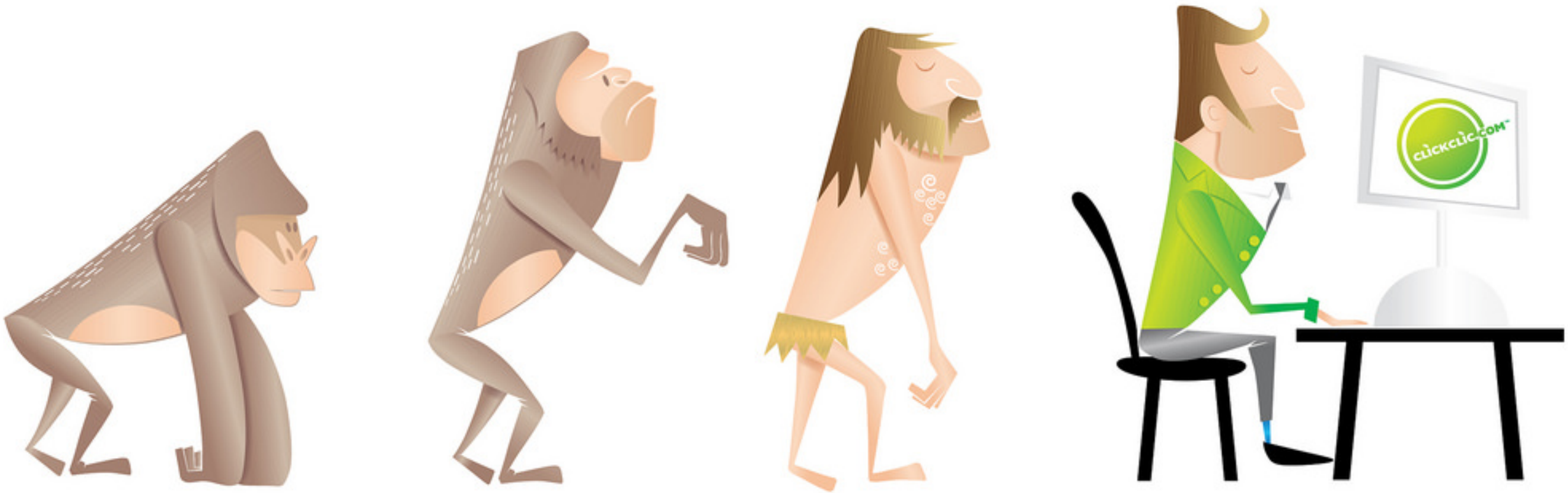


Storage

IaaS

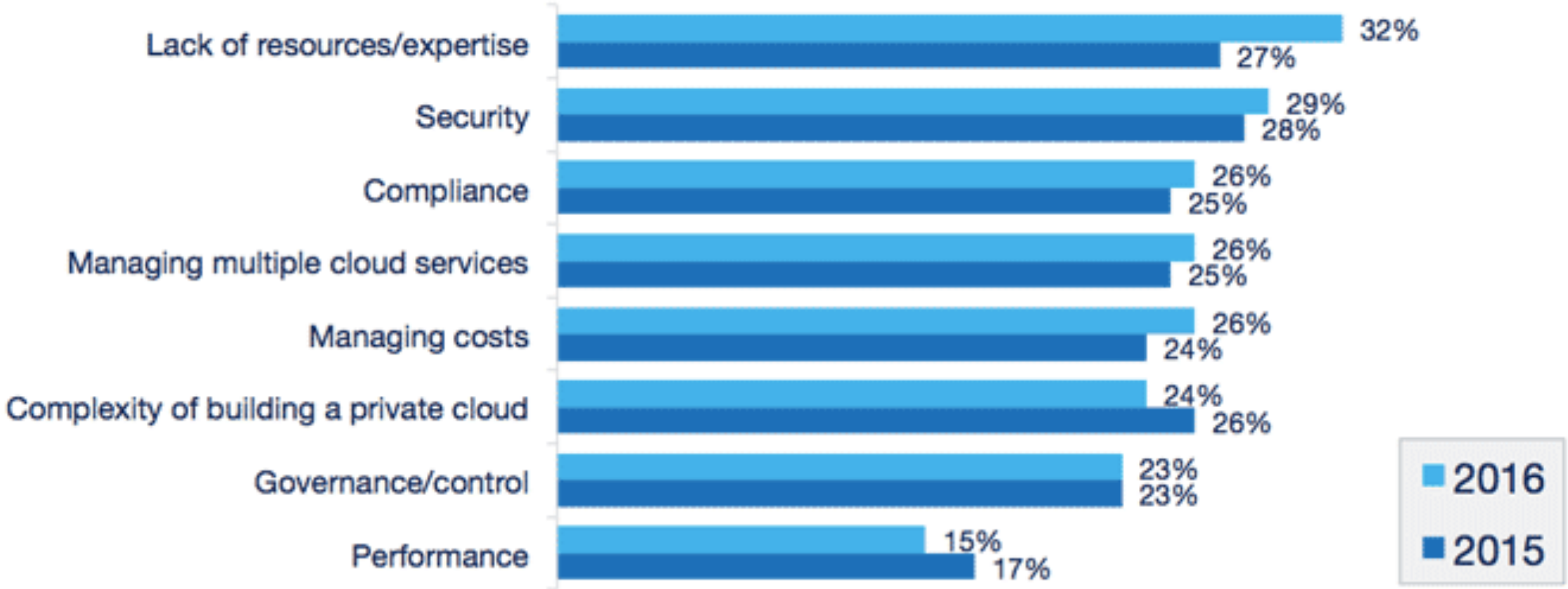
PaaS

FaaS

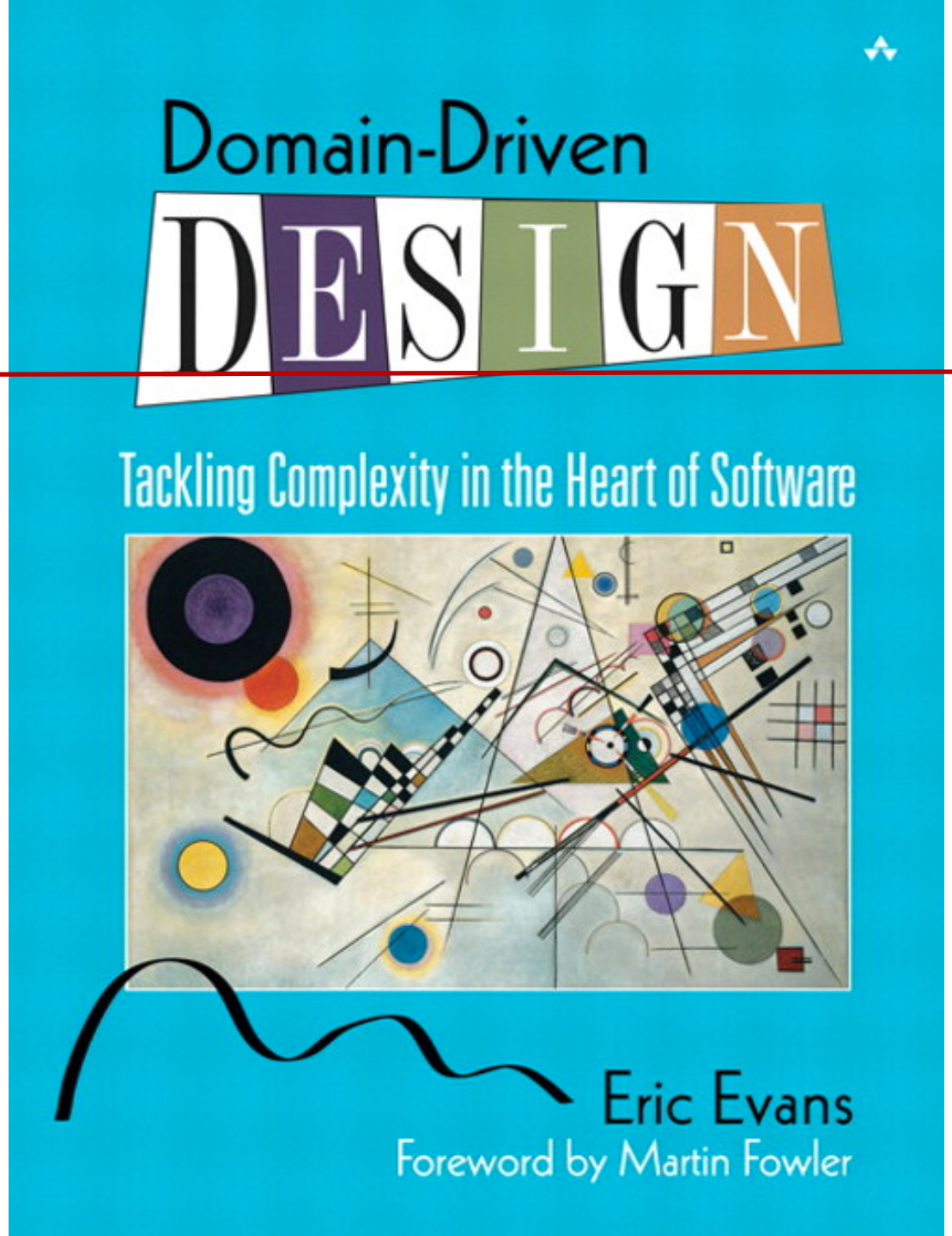

clickclic.com

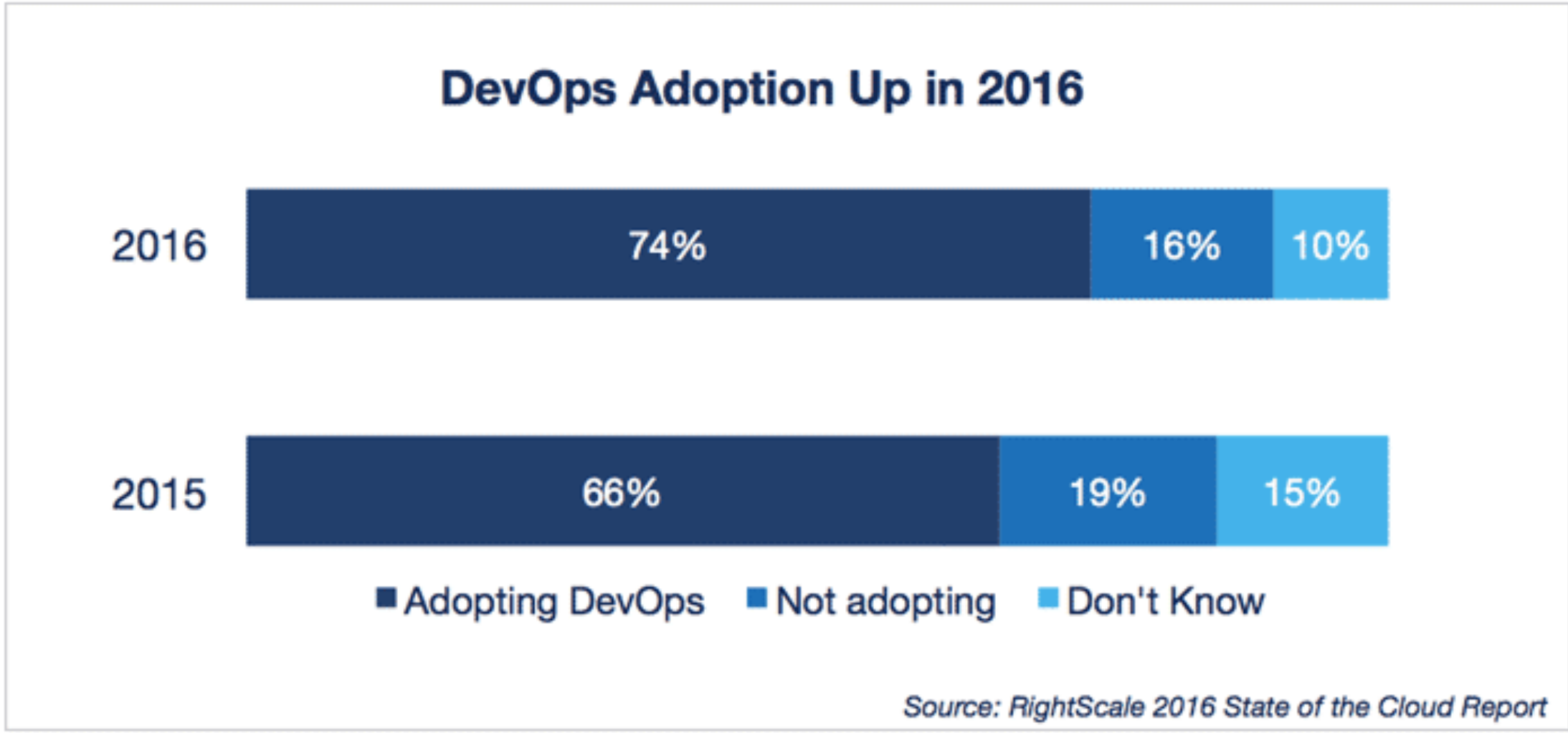

CLOUD FOUNDRY
puppet
elastic
ANSIBLE
docker
kubernetes
CHEF
Apache CASSANDRA

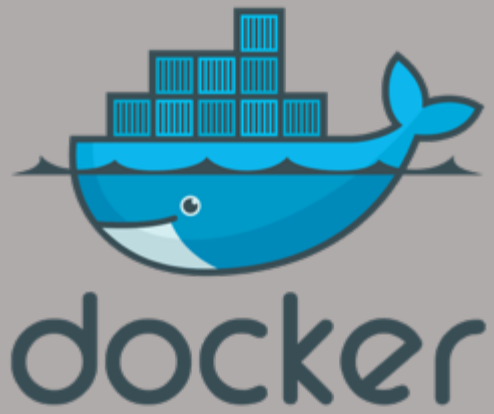
Cloud Challenges 2016 vs. 2015



Source: RightScale 2016 State of the Cloud Report


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler


DevOps Adoption Up in 2016
2016
74%
16%
10%
2015
66%
19%
15%
Adopting DevOps
Not adopting
Don't Know
Source: RightScale 2016 State of the Cloud Report


docker


puppet


CHEF™

microservices

“My First Law of Distributed Object Design: Don't distribute your objects ”

- Martin Fowler

final thoughts

- Don’t assume that you can do what you’ve done before.
- Invest in learning; innovation is happening quickly.
- Look for way to manage and reduce complexity.

- There is help:

  https://docs.microsoft.com/azure/guidance/

References

- *http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2016-state-cloud-survey*
- *http://www.idgconnect.com/view_abstract/34891/reach-clouds-enhanced-application-service-innovation-needs-flexible-dynamic-cloud-architecture-support*
- https://docs.microsoft.com/azure/security-center/
- https://azure.microsoft.com/regions/
- https://azure.microsoft.com/updates/
- https://azure.microsoft.com/campaigns/magic-quadrant/
- http://www.ijeit.com/vol%201/Issue%204/IJEIT1412201204_57.pdf

making choices

Choosing compute options

- Ease of migration
- Manageability
- Portability
- Programming model
- Level of density, isolation
- Cost


VM


VMSS


App Service


ACS


ServiceFabric


Azure Batch


Azure Functions

Choosing storage options

- SQL and NoSQL (KeyValue, Document, Column and Graph)
- Use managed services
- Go down to IaaS if knobs are not there
- Consider portability
- Schema on write/read
- Azure search for indexing

things that matter

- Scalability
- Availability
- Security
- Manageability (operations)
- Resiliency
- Cost

Scalability challenges

- Unbound scalability
  - Burst in requests, millions of devices
- Unpredictable hot spot
  - A single rock star can drain all resources
- Unexpected consistency/concurrency issue
  - ‘ACID’ mindset vs. relaxed consistency model
- Limited SKU
  - Can’t scale up infinitely
- Limits in each service
  - Scalability target
- Scale up/out incurs additional cost
  - N x instance = expenditure

Scalability best practices

- Decompose workload per different scalability requirement
- Partition data for better latency and throughput
- Use cache and CDN
- Default to asynchronous
- Relax consistency model
- Auto-scale to react to fluctuating load
- Avoid anti-patterns
- Perform load testing to detect all issues under peak load
- Monitor latency, throughput and errors in percentile all the time

Availability challenges

- Too much expectations
  - 99.999% = 26 secs of downtime a month
- Many failure points
  - HW failures, Service outage, Human errors
- No easy way for testing
  - Lack of fault injection test harness
- HA incurs additional cost
  - Redundancy = 2x expenditure

Availability best practices

- Deploy redundant services/components
- Deploy VMs in same tier to the same availability set
- Reduce downtime by blue/green deployment
- Backup/restore data
- Understand composite SLAs
- Implement resiliency strategy
- Perform fault injection testing
- Monitor latency, throughput and errors in percentile all the time

Security challenges

- Cyber attacks are common
  - DDoS, Malware injection, Side channel, Authentication, Man-in-the-middle
- Impact of data breach
  - Millions of PII have been leaking
- Authentication became a SPOF
  - One of the most critical components
- Conforming to compliance and regulations
  - PCI, SOX, HIPPA

Security best practices

- Configure NSG and/or NVA to protect resources from malicious users
- Use RBAC to restrict access to resources
- Consider federated authentication
- Perform threat modeling to identify and mitigate potential threats
- Grant least privilege permissions for resources
- Protect storage using shared access signatures
- Encrypt data using
- Manage secrets in KeyVault
- Mitigate DDoS attack at application layer

DevOps challenges

- Frequent update
  - Expectation for quicker TTM
- Monitoring and Root cause analysis
  - Are you all healthy? Who did this?
- Configuration
  - Snowflake servers
- Dependency management
  - Service, framework, library dependency

DevOps best practices

- Automate management tasks by scripting
- Use ARM template for provisioning
- Follow reliable release strategy
- Instrument applications
- Separate build, release and run concerns

$ Cost optimization

- Density is important
  - How much am I getting out of a single node? Vs how many nodes do I have?
  - Async I/O is often about improving density
  - Efficient use of a single resource (e.g. CPU)
  - Some sophisticated scheduling takes into account many types of resources (CPU, memory, disk, network)
- Choosing the “best fit” for the problem
  - Using SQL to parse XML isn’t cost effective
  - This often means managed services