StackEngine

Enterprise-Grade Docker Management

Open Containers of Champagne

OpenStack Magnum

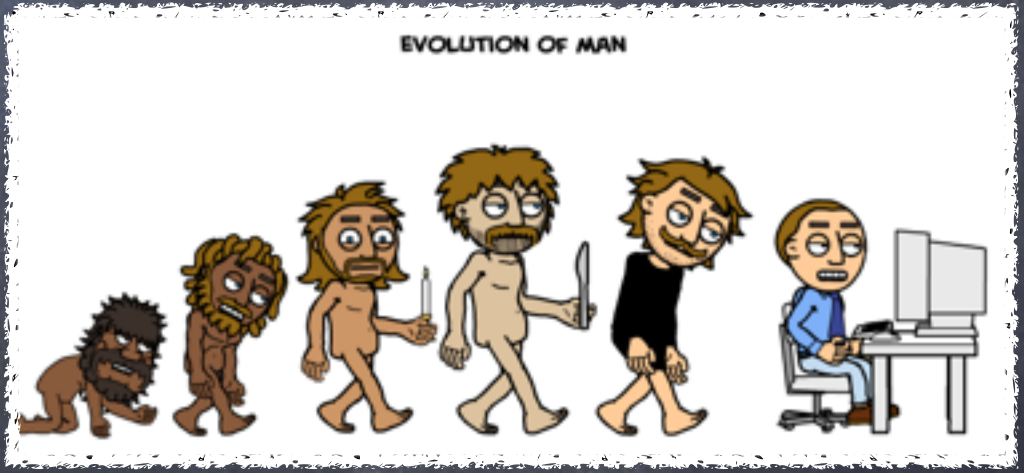
Boyd Hemphill
Director of Evangelism
@behemphi
@stackengine

Background

StackEngine

Background

- Oracle Developer, MySQL DBA, PHP Developer, System Architect, DevOps Director (snickering encouraged), Evangelist



StackEngine

Background

- Oracle Developer, MySQL DBA, PHP Developer, System Architect, DevOps Director (snickering encouraged), Evangelist
- Build Communities (Docker Austin, Austin DevOps)

StackEngine

Background

- Oracle Developer, MySQL DBA, PHP Developer, System Architect, DevOps Director (snickering encouraged), Evangelist
- Build Communities (Docker Austin, Austin DevOps)
- Founded Container Days 2015. Austin, Boston, San Fran, Dallas

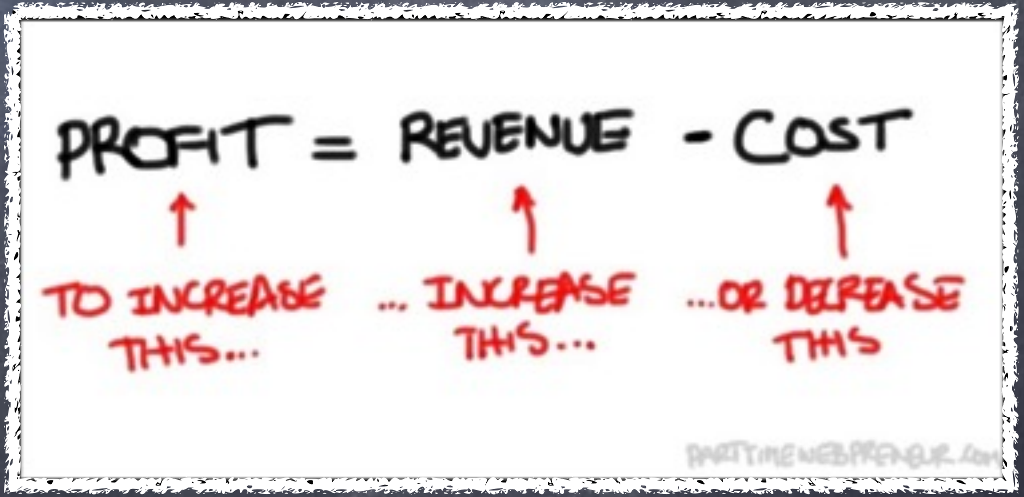
Why?

Walk away from this session thinking about uses for containers and what Magnum, Murano, nova-docker, et. al. could mean for the future of OpenStack.

Perspective

Perspective

- P = R - C



PROFIT = REVENUE - COST

TO INCREASE THIS... ... INCREASE THIS... ...OR DECREASE THIS

PARTTIMEWEBPRENEUR.COM

Perspective

- P = R - C
- If C = 0, you are out of business.

Perspective

- P = R - C
- If C = 0, you are out of business.
- R has no ceiling!

Perspective

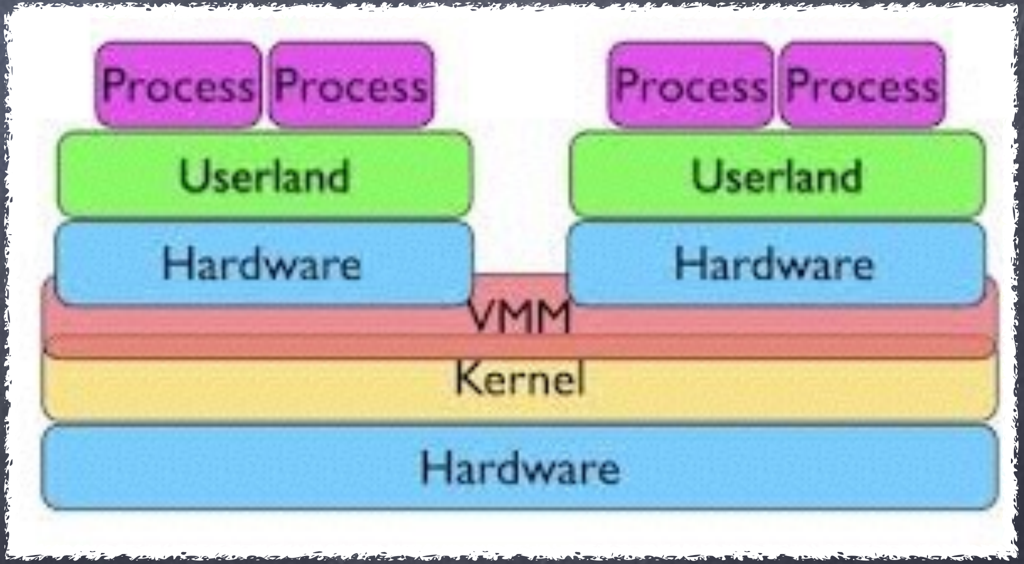
- P = R - C
- If C = 0, you are out of business.
- R has no ceiling!
- Features = $
    - http://goo.gl/VUmbwP
    - http://goo.gl/2Yq2Mg

Container v. VM

StackEngine

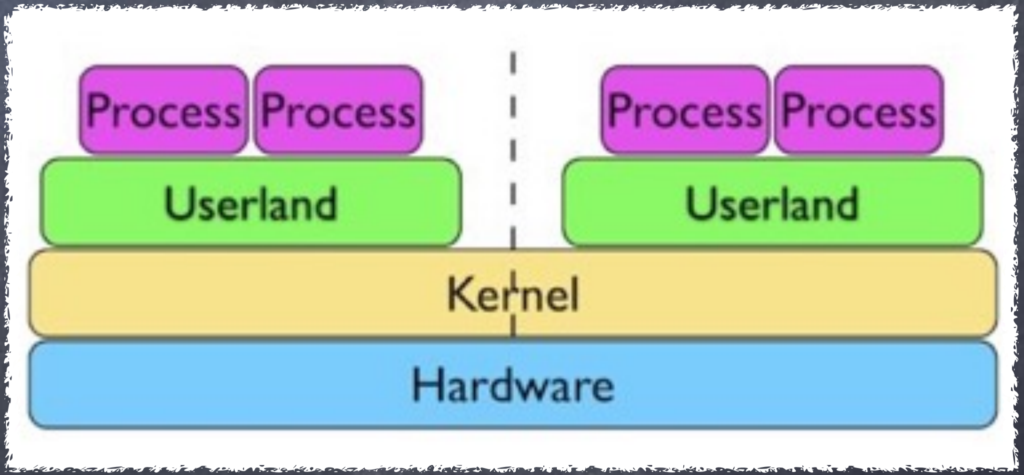
Container v. VM

- A VM is a full copy of on an entire computer running as software via a hypervisor



Container v. VM

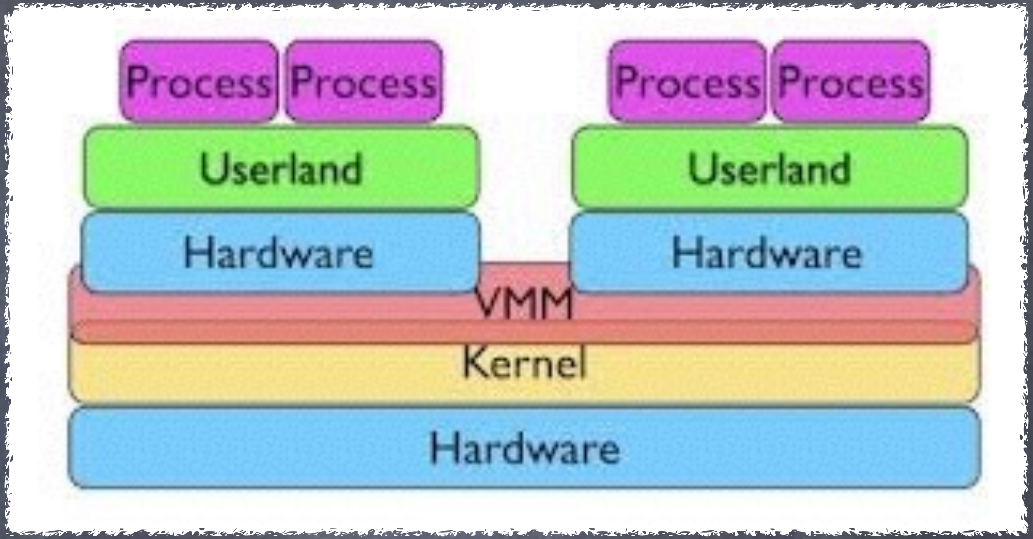
- A VM is a full copy of on an entire computer running as software via a hypervisor
- A container is a slice of the kernel

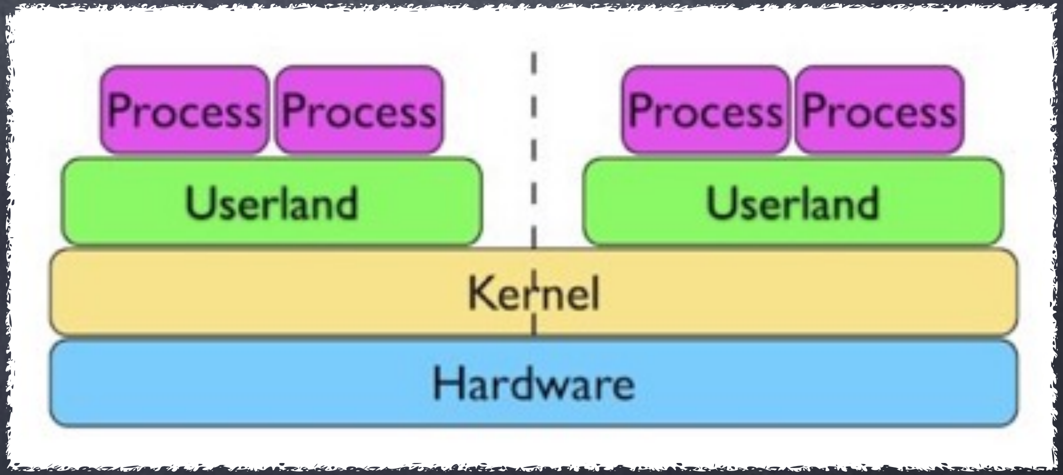


StackEngine

Container v. VM

- A VM is a full copy of on an entire computer running as software via a hypervisor
- A container is a slice of the kernel
- Executive Summary: The lack of extra layers means big opportunity





Language

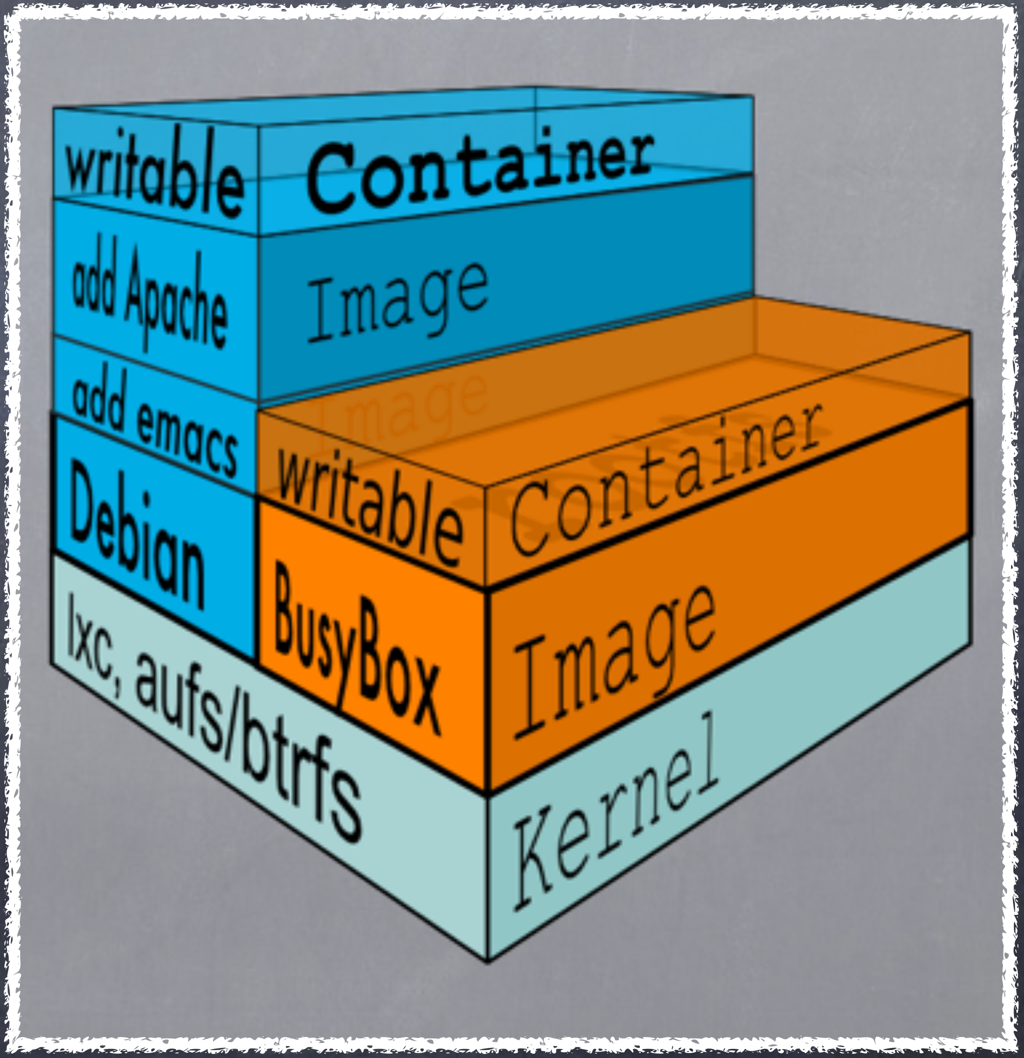
Docker != Containers

- FreeBSD Jail (1998) - CPU, Memory, Disk, !IO
- Solaris Zones (2005) CPU, Memory, Disk, IO
- OpenVZ (2005) CPU, Memory, Disk, IO
- LXC (2008) CPU, Memory, Disk, IO
- Docker (2013) CPU, Memory, !Disk, !IO
- Rocket (2014) ???

StackEngine

Docker != Container

- Containers in Production - Pantheon (LXC)



Docker != Container

- Containers in Production - Pantheon (LXC)
- Containers in build pipelines - Travis CI (OpenVZ)

Docker != Container

- Containers in Production - Pantheon (LXC)
- Containers in build pipelines - Travis CI (OpenVZ)
- Docker is, simply, linux containers for mere mortals

Magnum seeks to include container technologies other than Docker. Think broadly about the technology, not the tool.

Use Cases

StackEngine

Use Cases - SDLC

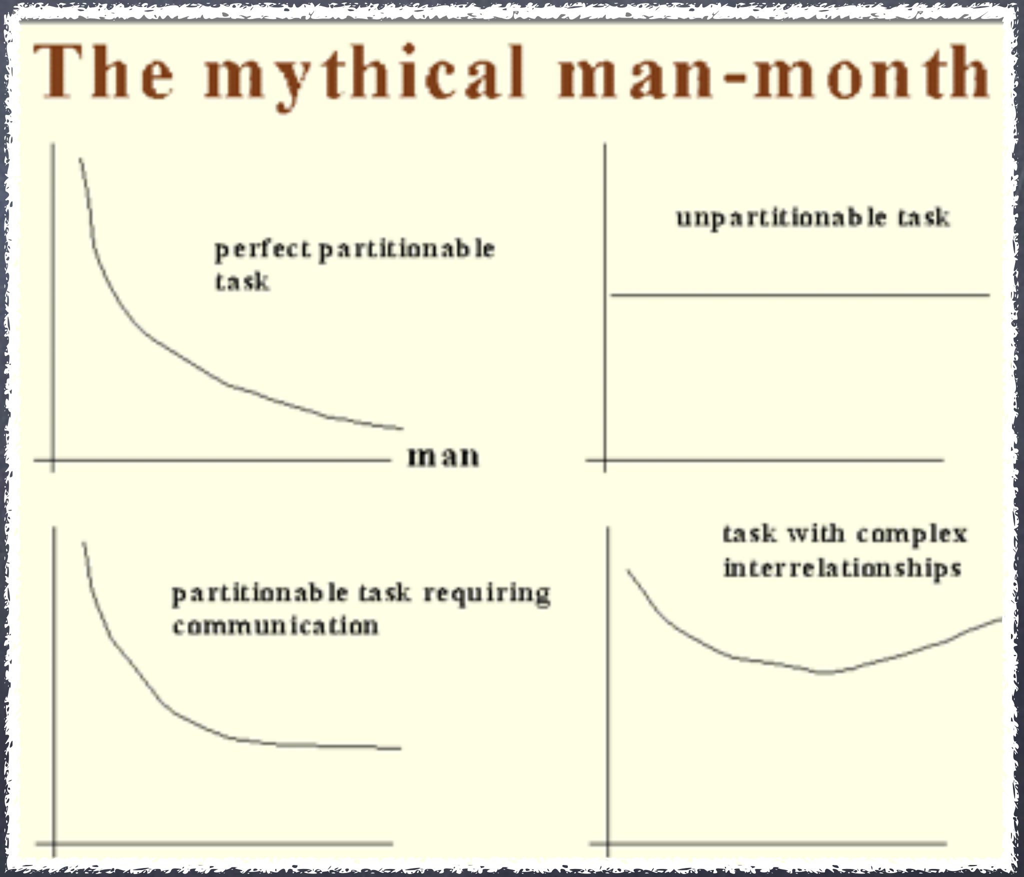
- Development
- Build / CI
- QA
- Production

Development

StackEngine

Use Case - Dev

- Disposable Dev Env
  - New Devs productive 1st day



StackEngine

Use Case - Dev

- Disposable Dev Env
  - New Devs productive 1st day
  - Innovation Increase

StackEngine

Use Case - Dev

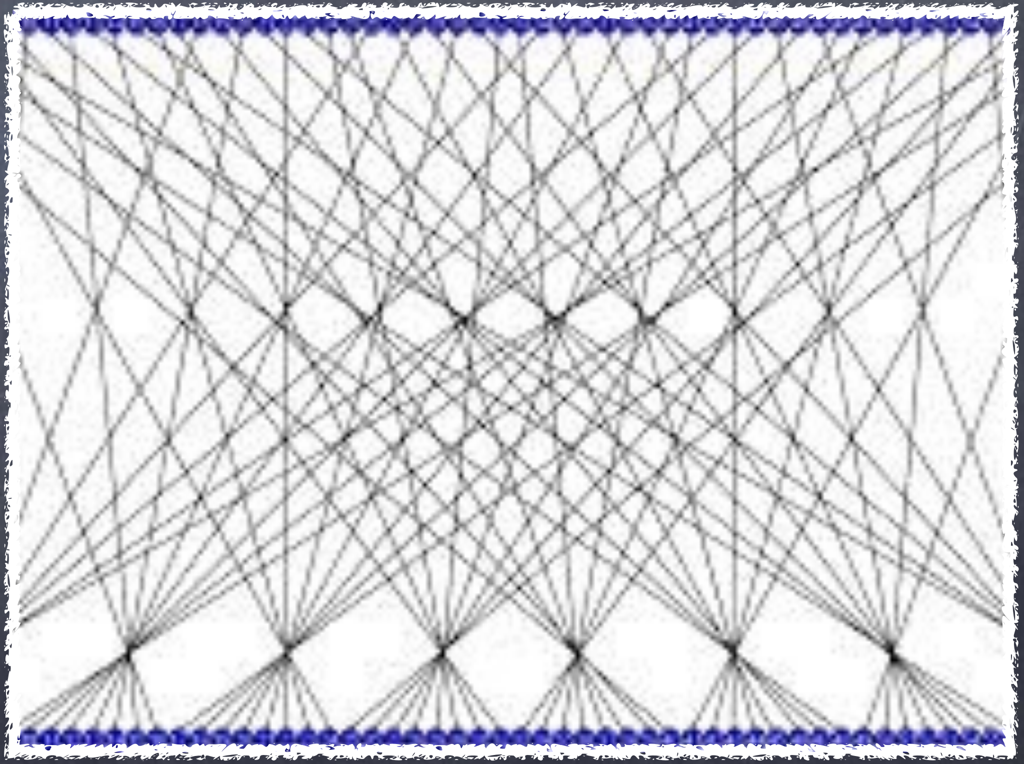
- Disposable Dev Env
  - New Devs productive 1st day
  - Innovation Increase
  - Feature Velocity Increase

Build / CI

StackEngine

Build / CI

- Test more system permutations



Build / CI

- Test more system permutations
- Increased parallelism = increased innovation

StackEngine

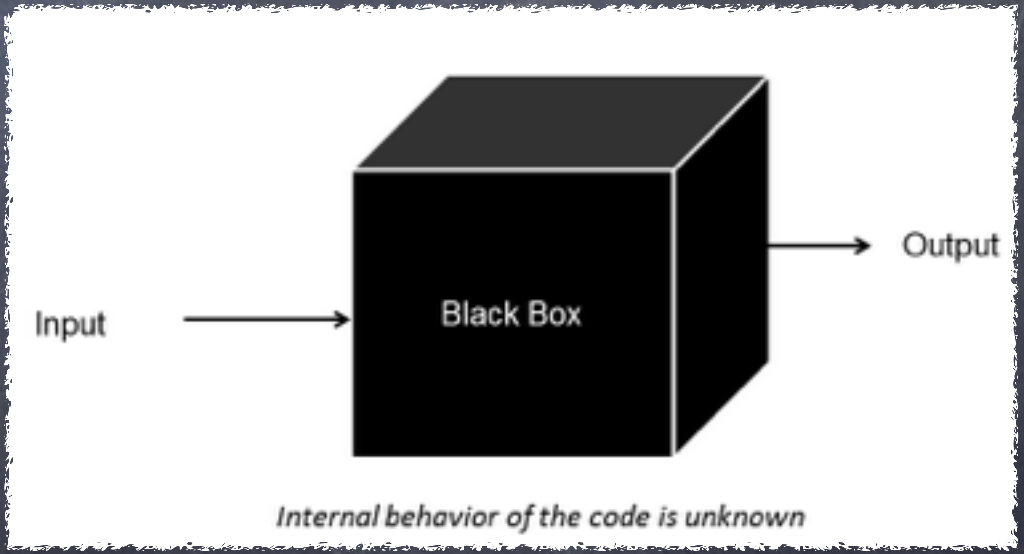
Build / CI

- Test more system permutations
- Increased parallelism = increased innovation
- Increased parallelism = increased feature velocity

Quality Assurance

Quality Assurance

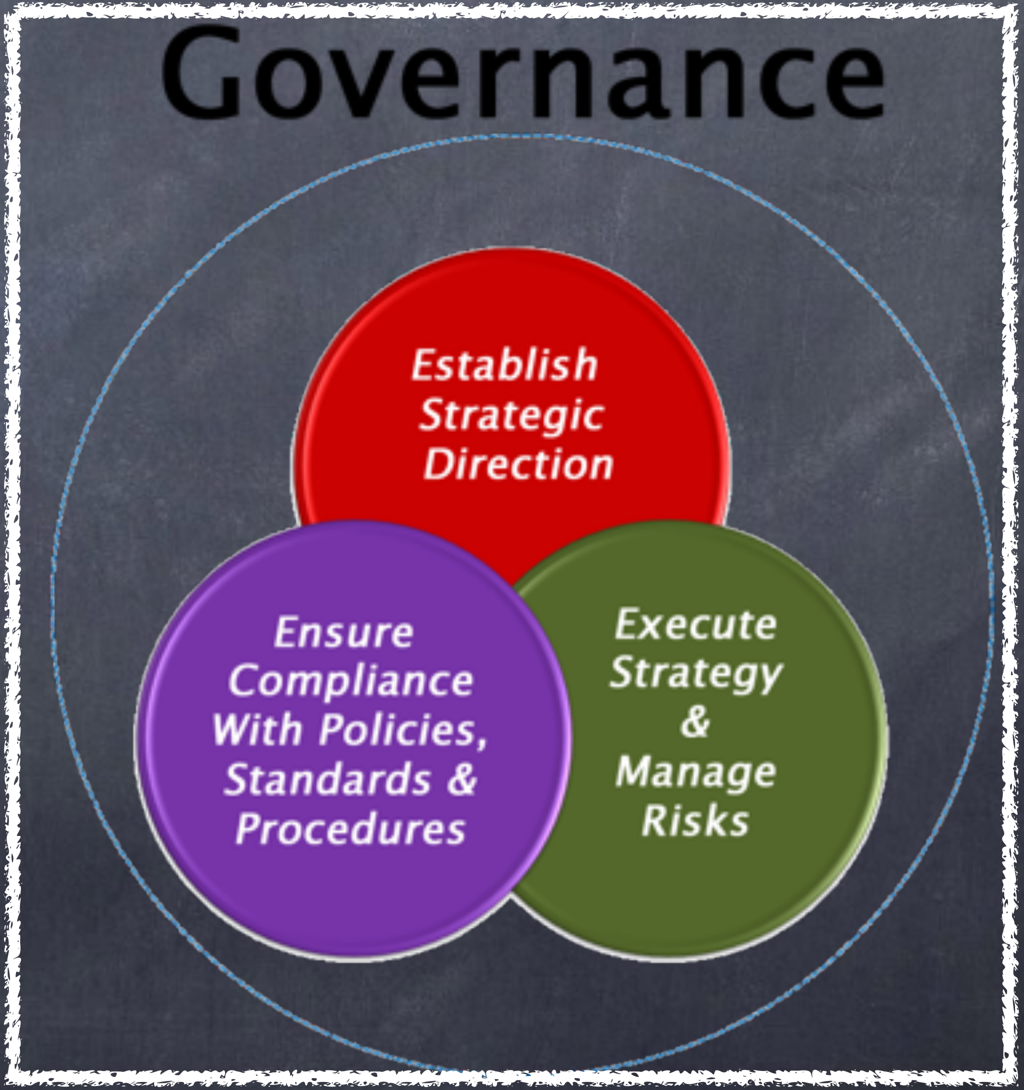
- Black Box Testing



StackEngine

Quality Assurance

- Black Box Testing
- Compliance instead of Governance
  - Security
  - Performance
  - Functional


Governance
Establish Strategic Direction
Ensure Compliance With Policies, Standards & Procedures
Execute Strategy & Manage Risks

StackEngine

Quality Assurance

- Black Box Testing
- Compliance instead of Governance
  - Security
  - Performance
  - Functional
- Increases Feature Velocity

Operations

Operations

- Process Density

StackEngine

Operations

- Process Density
- Ants not Cattle

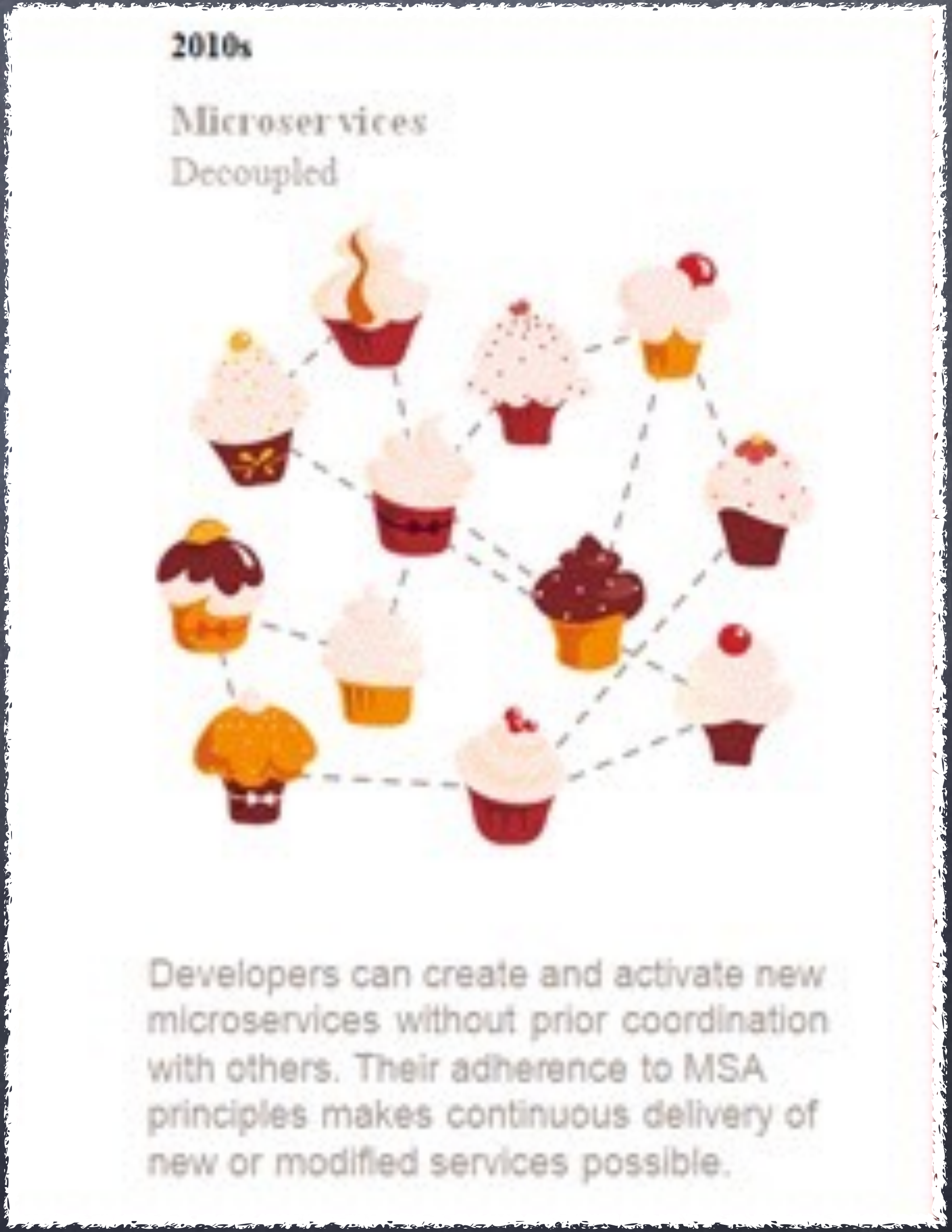
Operations

- Process Density
- Ants not Cattle
- New Capacity Planning thinking

Bonus

System Architecture

- Micro Services mean Micro teams



2010s

Microser vices
Decoupled

Developers can create and activate new microservices without prior coordination with others. Their adherence to MSA principles makes continuous delivery of new or modified services possible.

StackEngine

System Architecture

- Micro Services mean Micro teams

System Architecture

- Micro Services mean Micro teams
- Increased Feature Velocity
- Lasting competitive business advantage

Batteries Not Included

Batteries Not Included

- Docker ~= Hypervisor and not much else

StackEngine

Batteries Not Included

- Docker ~= Hypervisor and not much else
- Further symmetry
  - !Storage (Swift, Cinder)
  - !Networking (Neutron)
  - !Identity (Keystone)

StackEngine

Batteries Not Included

- Docker ~= Hypervisor and not much else
- Further symmetry
  - !Storage (Swift, Cinder)
  - !Networking (Neutron)
  - !Identity (Keystone)
- Nascent Ecosystem

OpenStack, via Magnum, has the potential to provide excellent answers to very difficult questions about container workloads and application topologies

Parting Thoughts

StackEngine

- Developer adoption of Docker is only valuable as a first step. There is not enough benefit from it alone to justify the effort, it must inform system architecture and production operations <u>over time.</u>

- Developer adoption of Docker is only valuable as a first step. There is not enough benefit from it alone to justify the effort, it must inform system architecture and production operations over time.
- Docker's system architecture ramifications have the potential to provide a significant and lasting competitive business advantage

StackEngine

- Developer adoption of Docker is only valuable as a first step. There is not enough benefit from it alone to justify the effort, it must inform system architecture and production operations over time.

- Docker's system architecture ramifications have the potential to provide a significant and lasting competitive business advantage

- Unlike most improvements from DevOps thinking, container adoption is Dev/QA driven since the greatest benefits are from system architecture. This fits existing common OpenStack use cases.

StackEngine

- Developer adoption of Docker is only valuable as a first step. There is not enough benefit from it alone to justify the effort, it must inform system architecture and production operations over time.
- Docker's system architecture ramifications have the potential to provide a significant and lasting competitive business advantage
- Unlike most improvements from DevOps thinking, container adoption is Dev/QA driven since the greatest benefits are from system architecture. This fits existing common OpenStack use cases.
- Horses are starting to become unicorns. Evolve or die.

StackEngine

- Developer adoption of Docker is only valuable as a first step. There is not enough benefit from it alone to justify the effort, it must inform system architecture and production operations over time.
- Docker's system architecture ramifications have the potential to provide a significant and lasting competitive business advantage
- Unlike most improvements from DevOps thinking, container adoption is Dev/QA driven since the greatest benefits are from system architecture. This fits existing common OpenStack use cases.
- Horses are starting to become unicorns. Evolve or die.
- The last point is FUD. "Consider how to evolve or die," is the real truth.

Colophon

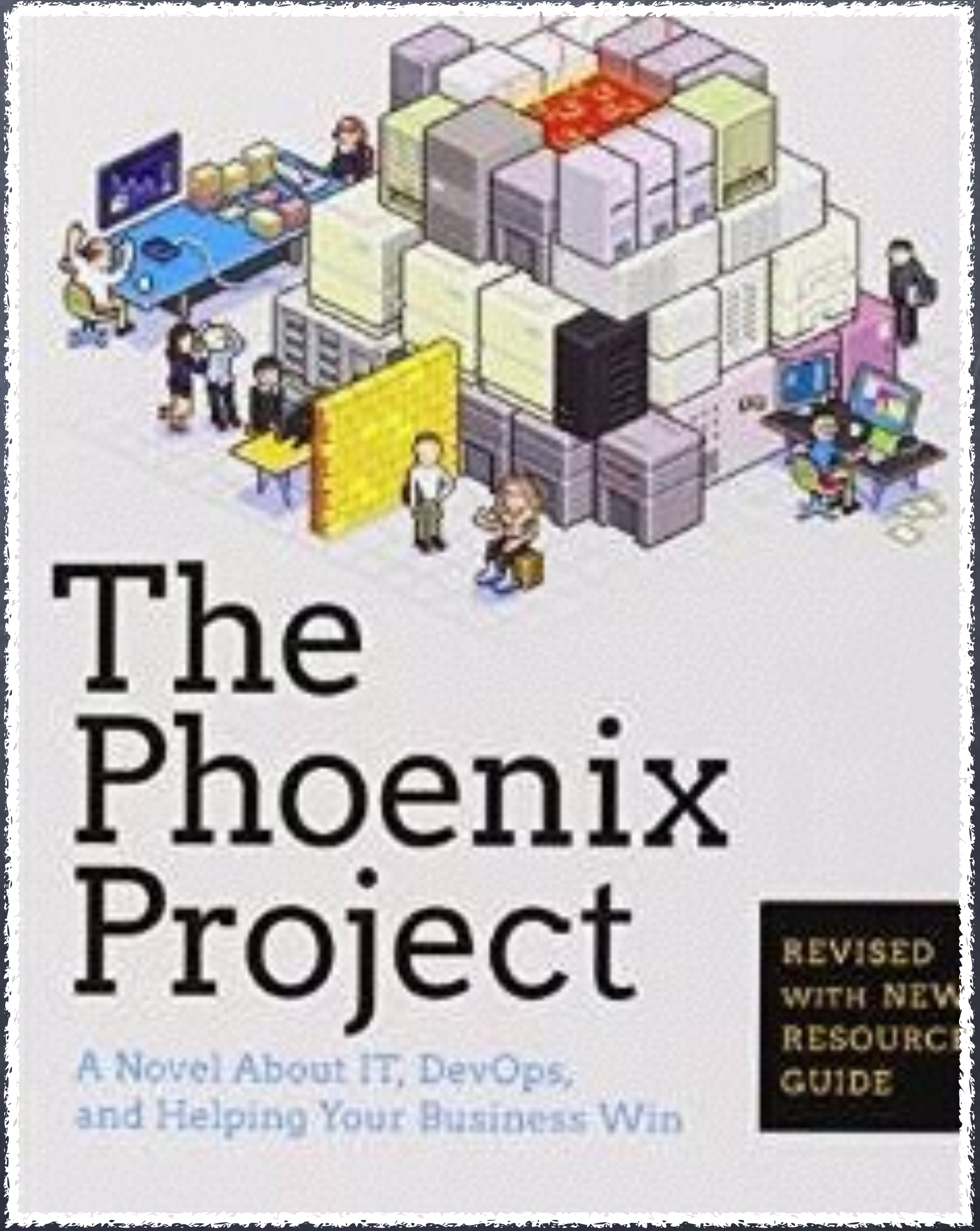
Unicorn Case

- Pantheon/Rackspace blog: http://goo.gl/93steF
- Note the date: October 7th, 2013 (This is LXC)
- Amazing video presentation: http://goo.gl/gRkKGN
- Docker makes this easier to consider in our own contexts

StackEngine

Reading

- The Phoenix Project - Kim (Lean/tech)
- The Lean Startup - Reis (Lean)
- The Goal - Goldratt (Theory of Constraints)
- It's not Luck - Goldratt (Theory of Constraints)
- Good to Great - Collins (Culture)
- The No Asshole Rule - Sutton (Culture)
- Continuous Deliver - Humble (*) (deep tech)
- The Lean Enterprise - Humble (*) (Lean)
- Twelve Factor - http://12factor.net (deep tech)


The Phoenix Project
A Novel About IT, DevOps, and Helping Your Business Win
REVISED WITH NEW RESOURCE GUIDE