Bringing DevOps to Everyone

—

Bernard Sanders
CTO, CloudBolt Software

Agenda - Bringing DevOps to Everyone

Part I - The Background

i. The State of IT Today
ii. The Goal
iii. History of DevOps

Part II - How to do it

i. The social & organizational prerequisites
ii. The 4-layer model for a shared DevOps Practice

Part I - The Background

The Current State of IT Departments

Expressed in quotes

- "It takes us 6 months to build VM" - a major bank
- "It takes us 29 steps to build a VM" - a major manufacturing conglomerate
- "We may be able to get that in DNS by the end of the week" -a tech company
  - (they couldn't)
- "It's hard to get a [private] IP address" -a premium brand retailer
- "I don't know which of these VM templates might work" -a state government
- "We don't know who these VMs belong to, or whether they're still needed" - most IT orgs

The Perception of IT Orgs

Do not, my friends, become addicted to water [servers]. It will take hold of you, and you will resent its absence!

-Immortan Joe

The Reality

- Change can be very hard
  - cultures of status quo
  - it can also takes a lot of work
- Most IT organizations have been bitten
- Risk is not rewarded

A Brief History of DevOps

In general categories:

- **Basic automation:** 1990s
  - **Cfengine:** 1993
- **Datacenter Automation:** Opsware / Bladelogic - 2000-2007
- **Config Management:** Puppet / Chef ~2004-
- **DevOps:** coined in 2007

Key takeaways:

- Trying to share the power of automation is not new
- The tools necessary to actually accomplish this are just starting to mature

See also: Rachel Chalmers talk on the history of automation

The Goal: Bring DevOps to Everyone

- Anyone who needs any IT assets, internally or externally, should be able to easily provision, manage, control, and eventually/automatically remove them
- Anyone: developers & operations, but also support, QA, marketing, sales, and management
- The costs and other trade-offs should be clearly presented to them in a clean & simple UI
- DevOps goal should be SDIT - Software Defined IT

Why Should I?

If you have some sweet automation, but only a few people can use it, you are not realizing your full potential

The Benefits to Developers

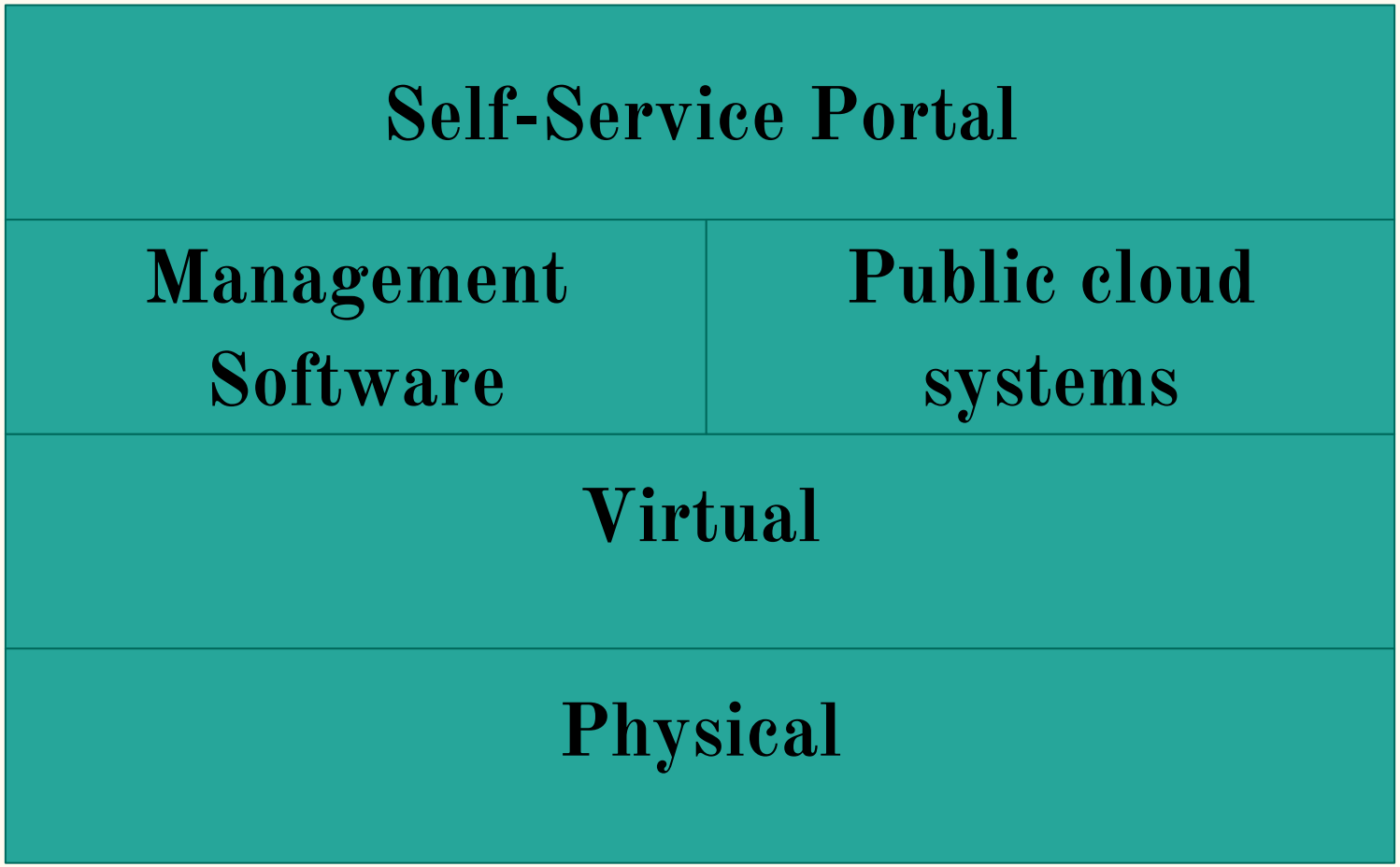
- Stop worrying about quotidian operational details
- Consistent, rapid deployment
- Continuous infrastructure testing
- Sprawl avoidance – so you don't run out of resources
- Snapshotting – enabling quick replication/rollback
- Ease of capacity reconfiguration – allowing developers to quickly pivot from prototyping to stress testing

Part II - How to do it

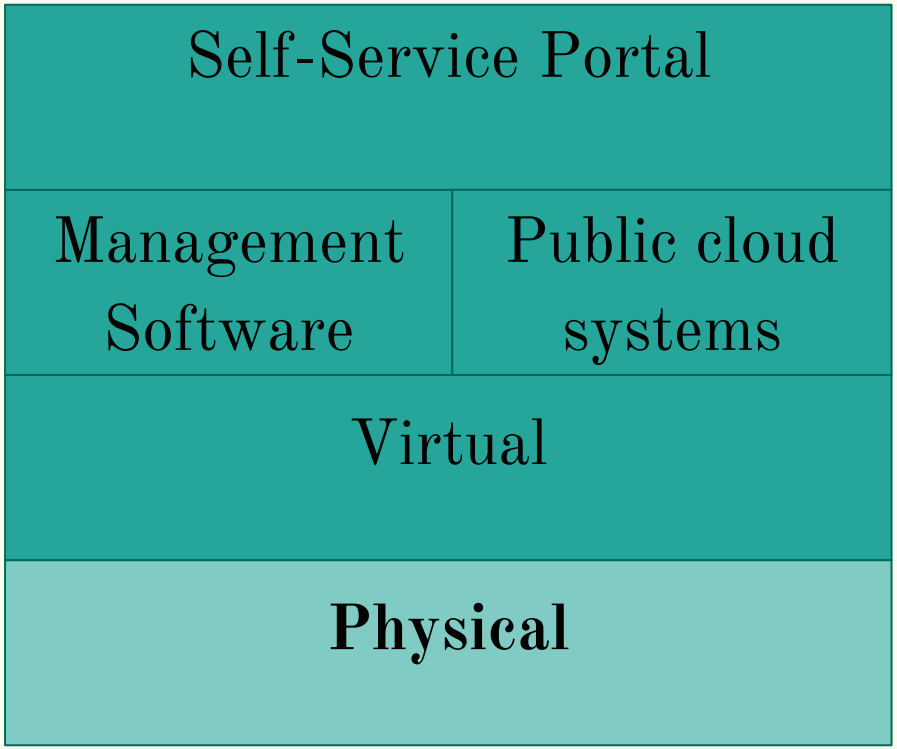
The Social & Organizational Prerequisites

- A culture that celebrates change *(but does it carefully)*
- A shared vision
- Cross-organizational trust
- A desire in IT to make other teams successful
- A high standard for IT delivery set by management
- Tons of communication
  - a common, shared knowledge store (Google Drive, wiki, etc)
  - a messaging system (Slack, HipChat, IRC, etc)
  - easy to use screensharing that anyone can start anytime
  - periodic, scheduled live knowledge sharing sessions

The 4-Layer Model for a Shared DevOps Practice

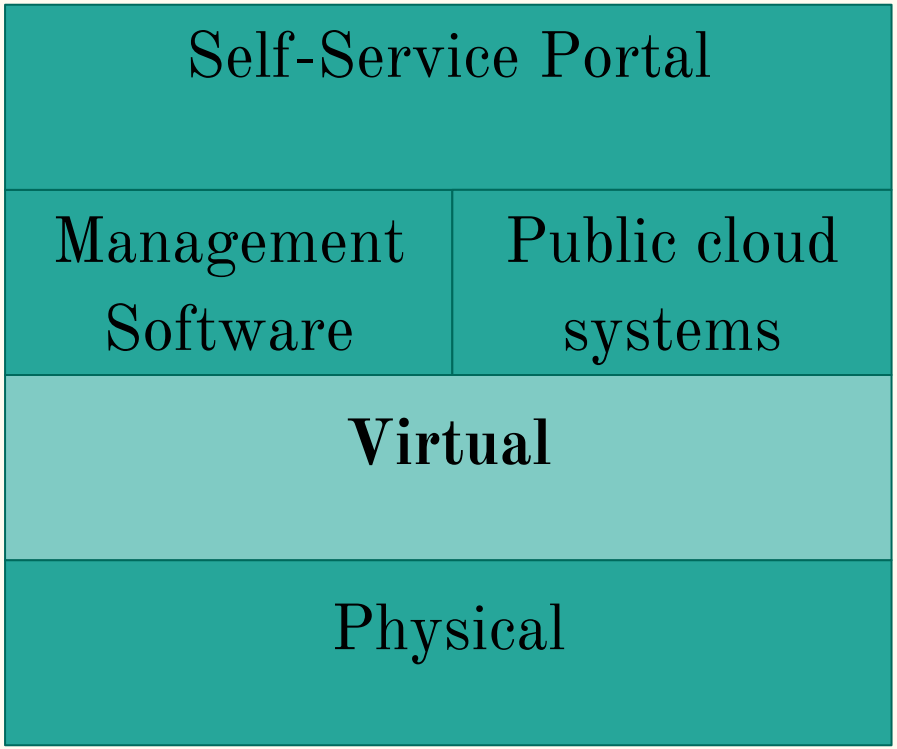


The 4-Layer Model: **Physical**



Servers, storage devices, network gear, cables

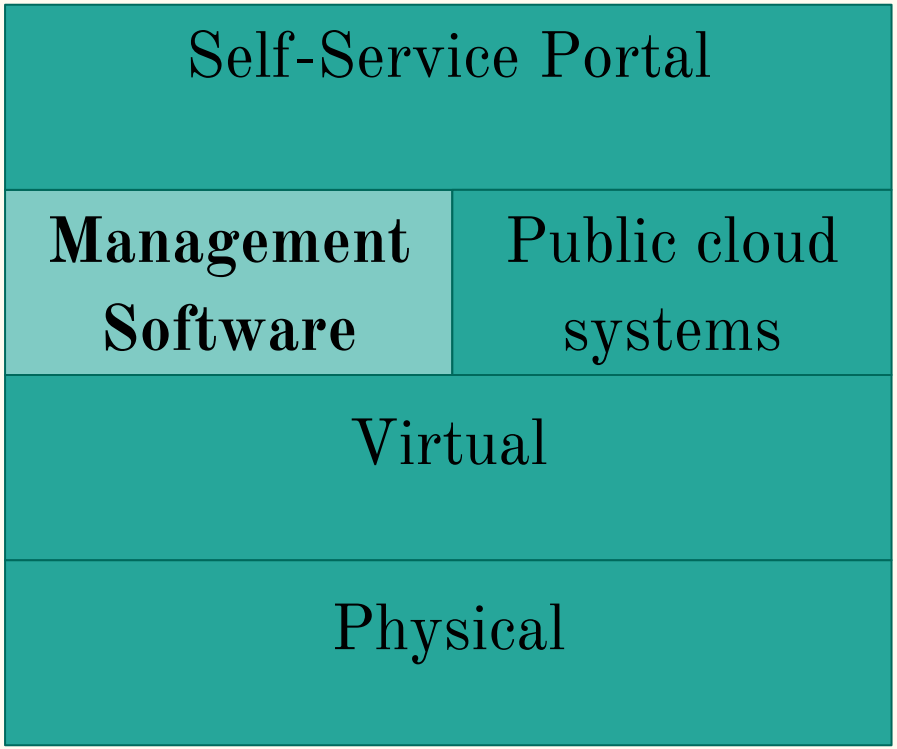
The 4-Layer Model: **Virtual**



Hypervisors

Examples:
VMware ESX, KVM, XenServer

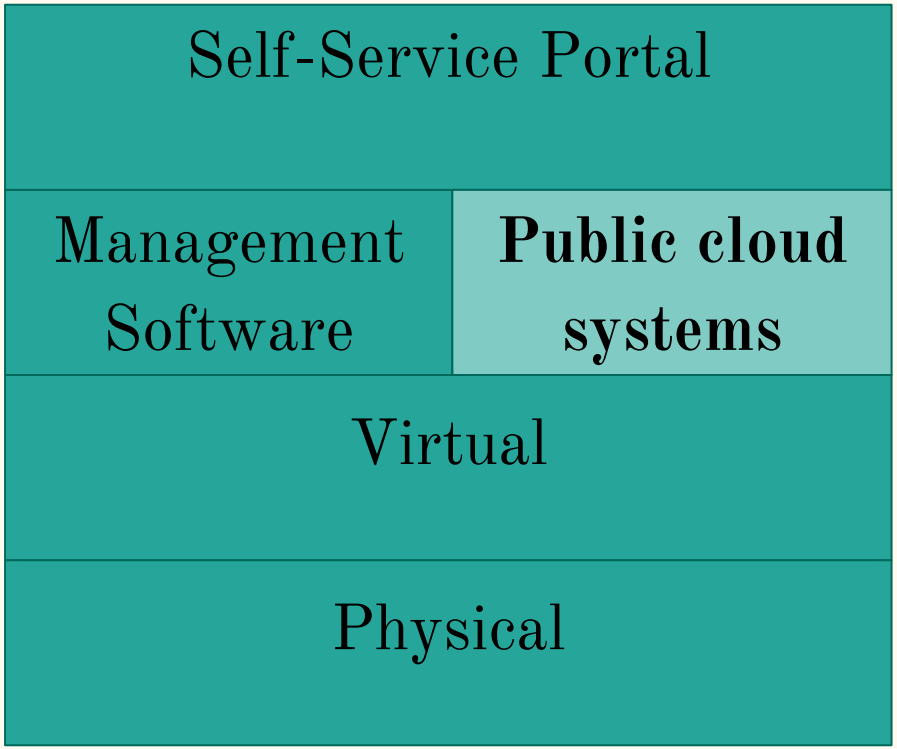
The 4-Layer Model: **Management Software**



Sub-categories:

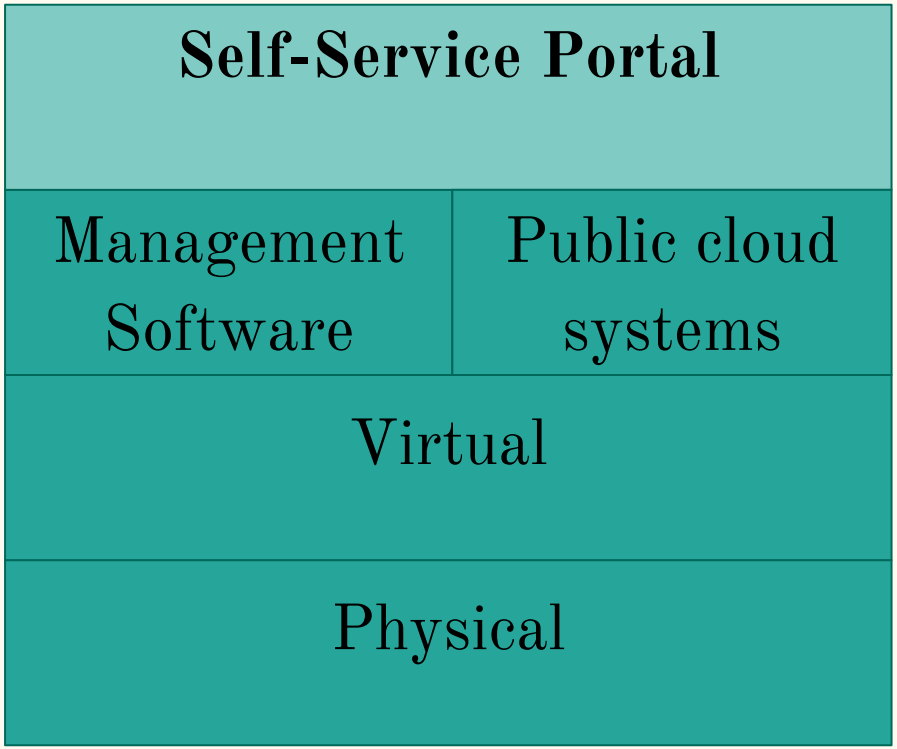
- **Hypervisor Mgmt-** *vCenter, OpenStack*
- **Config Mgmt-** *Puppet, Chef, Ansible, Salt*
- **Monitoring**
- **Backup**
- **IP/DNS/DHCP management**
- **Runbook automation**
- **Bare metal provisioning**
- **CMDB**
- **Ticketing**

The 4-Layer Model: **Public Cloud Systems**



Examples: AWS, Azure, CTL cloud, Digital Ocean, Linode, Oracle Public Cloud, Rackspace, Softlayer, etc

The 4-Layer Model: **Self-Service Portal**



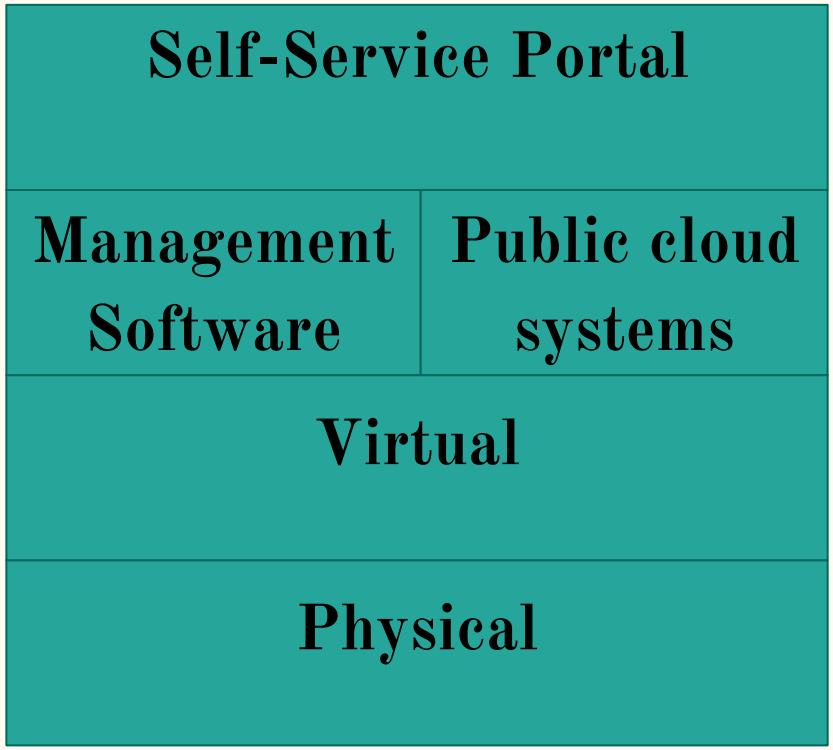
Examples: CliQr, CloudBolt, Embotics, HP CSA, RedHat CloudForms, RightScale, VMware vRA


And then what happened?
CalmingManatee.com
ROFLBOT

Go Up the Stack

Self-Service Portal	
Management Software	Public cloud systems
Virtual	
Physical	

Critical, but overlooked, product eval factors



- Weight of the solution
- Self-controlled extensibility
- Integrability
- Technology-agnostic
- Can it work resources you already have? (brownfield)
- Delightfulness of user experience
  - If it's not simple & delightful to use, people won't

Bernard Sanders CTO, CloudBolt Software

bernard@cloudbolt.io http://cloudbolt.io/