Cloud Computing

A little history . . . Life in the Corporate “Data Center”

A little history . . .

- Things to think about:
  - Scaling, Lead time
  - Floor space
  - Power management
  - Heating/Cooling
  - Redundancy -- SPOF
  - Fire Suppression
  - Battery Backup -- UPS
  - Network Wiring
  - Data & Server Backups
  - Network Switching
  - 7X24 support
  - Alerts/Alarms

Welcome, "Cloud Computing"

- Private – my private cloud in my own data center
- Public – a shared environment hosted by a provider

A metaphor

- The electrical grid
  - You don't know where it comes from
  - It's there when you need it, just plug it in
  - Use what you want
  - Need more? Just take it.
  - Pay for what you use

Cloud Computing

Definition . . .

- **"Computing Services and Solutions are delivered and consumed in real time over the internet."**

- **Characteristics of Public Cloud Services**
  - Offsite hosting
  - Pay per use (setup/initial, plus ongoing)
  - Shared space
  - Massively Scalable
  - On-Demand Provisioning
  - Rapid Deployment
  - Lowers innovation barriers
  - Leading edge architecture

Advantages of Cloud Computing

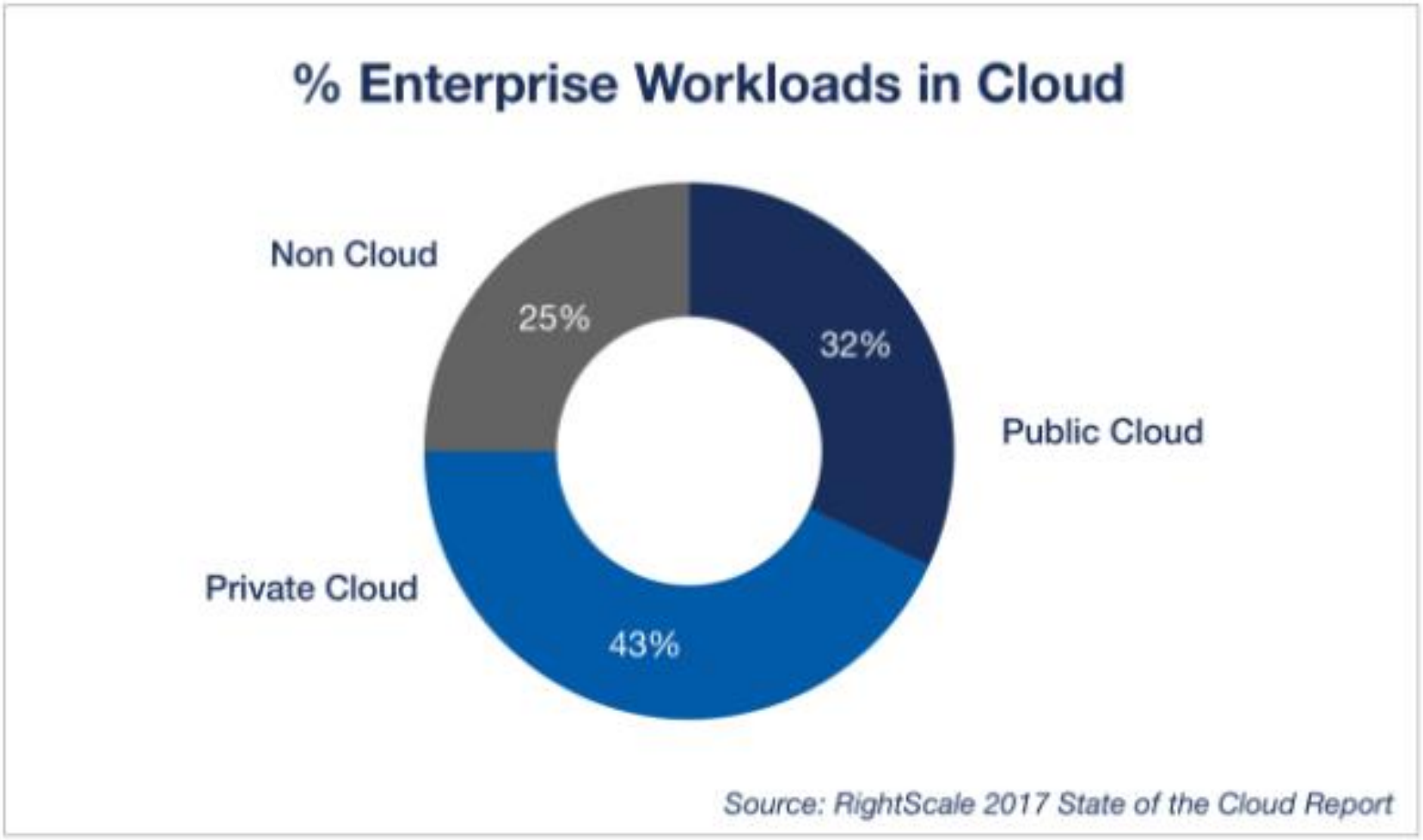
- Ubiquitous (available from anywhere)
- Automated change management
- Massively Scalable
- On-Demand Provisioning
- Rapid Deployment
- Lowers innovation barriers
- Leading edge architecture
- Lower Cost

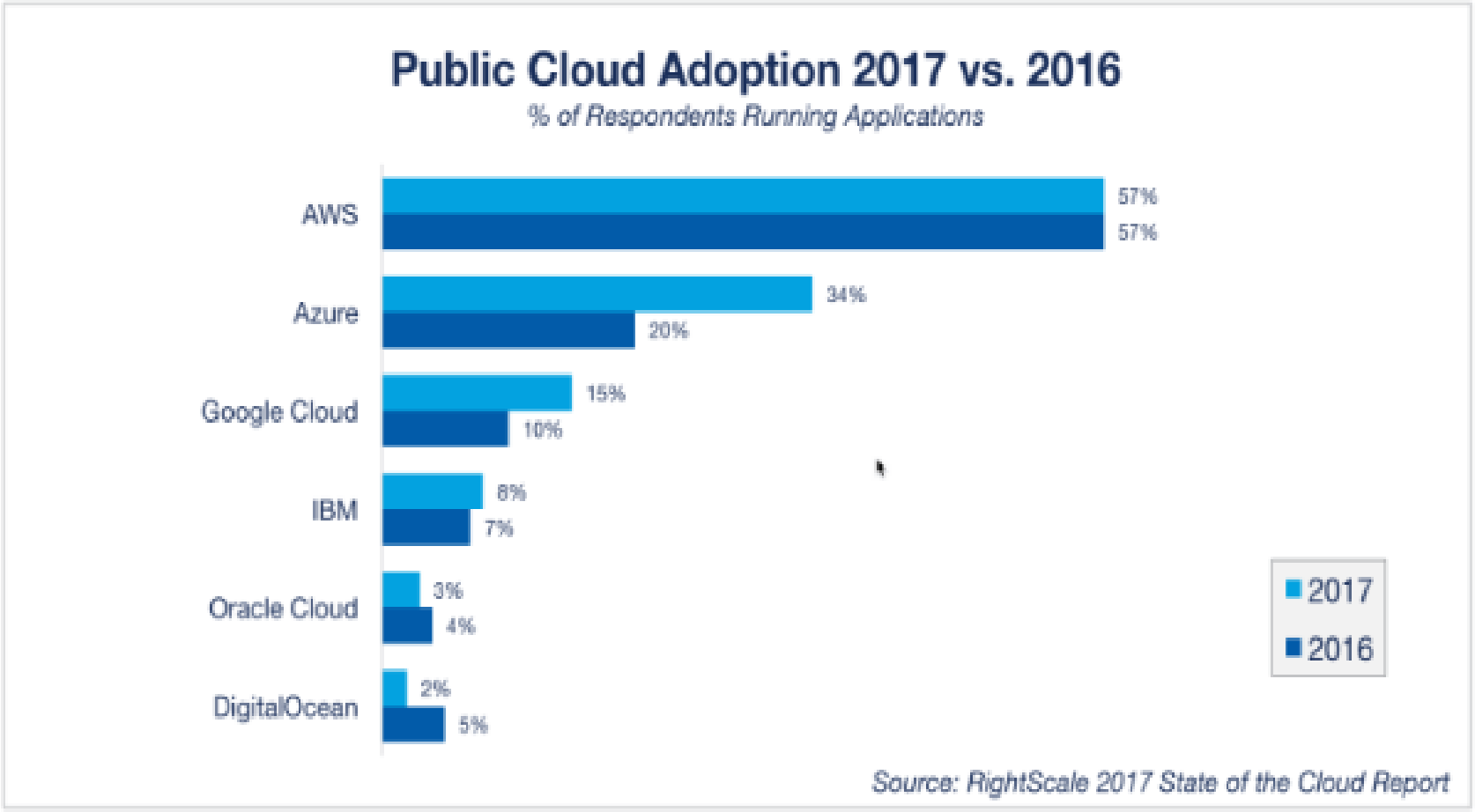
Disadvantages of Cloud Computing

- Surrender Control
- Less Robust Monitoring
- Requires Large Network Pipe/Capacity
- Less Secure – multi-tenancy, DDOS

Private versus Public Cloud

- Private: Leverage the advantages, with few disadvantages
  - Massively Scalable
  - On-Demand Provisioning
  - Rapid Deployment
  - More secure
  - Better Monitoring
  - BUT → Still requires significant internal infrastructure


% Enterprise Workloads in Cloud
Non Cloud
25%
32%
Public Cloud
Private Cloud
43%
Source: RightScale 2017 State of the Cloud Report

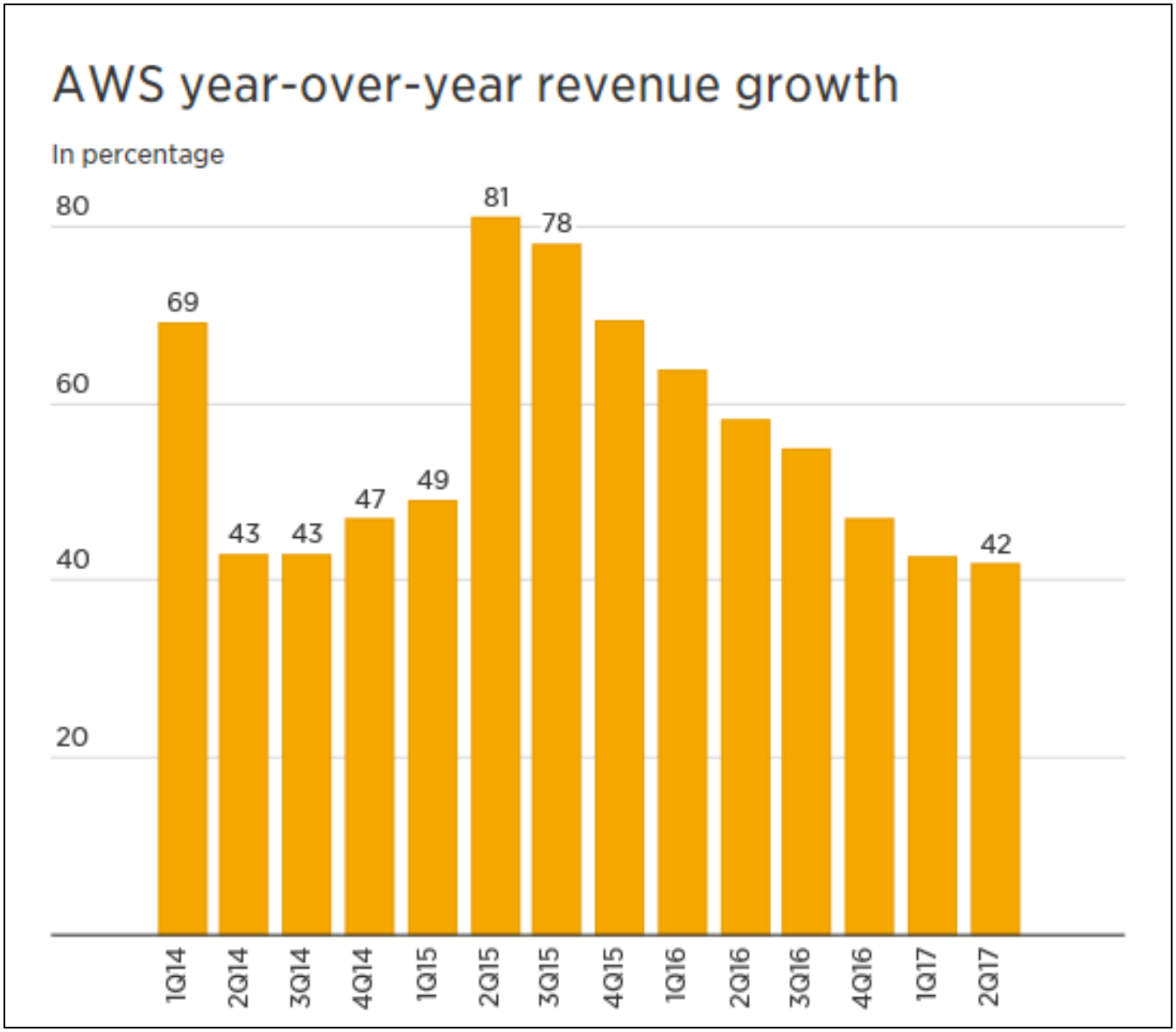

Public Cloud Adoption 2017 vs. 2016
% of Respondents Running Applications
AWS
57%
57%
Azure
34%
20%
Google Cloud
15%
10%
IBM
8%
7%
Oracle Cloud
3%
4%
DigitalOcean
2%
5%
2017
2016
Source: RightScale 2017 State of the Cloud Report

Updates from IDC

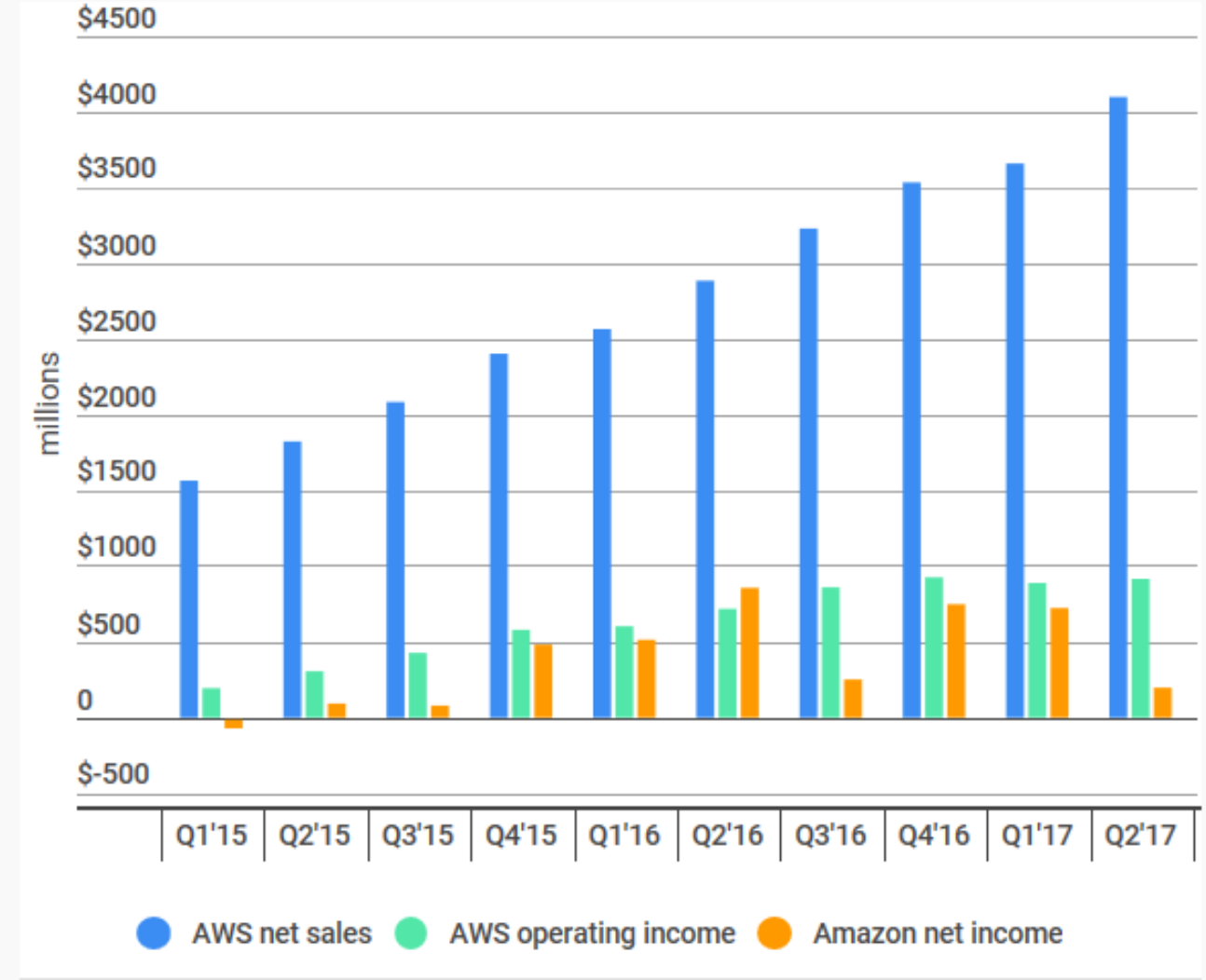
(The premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets.)

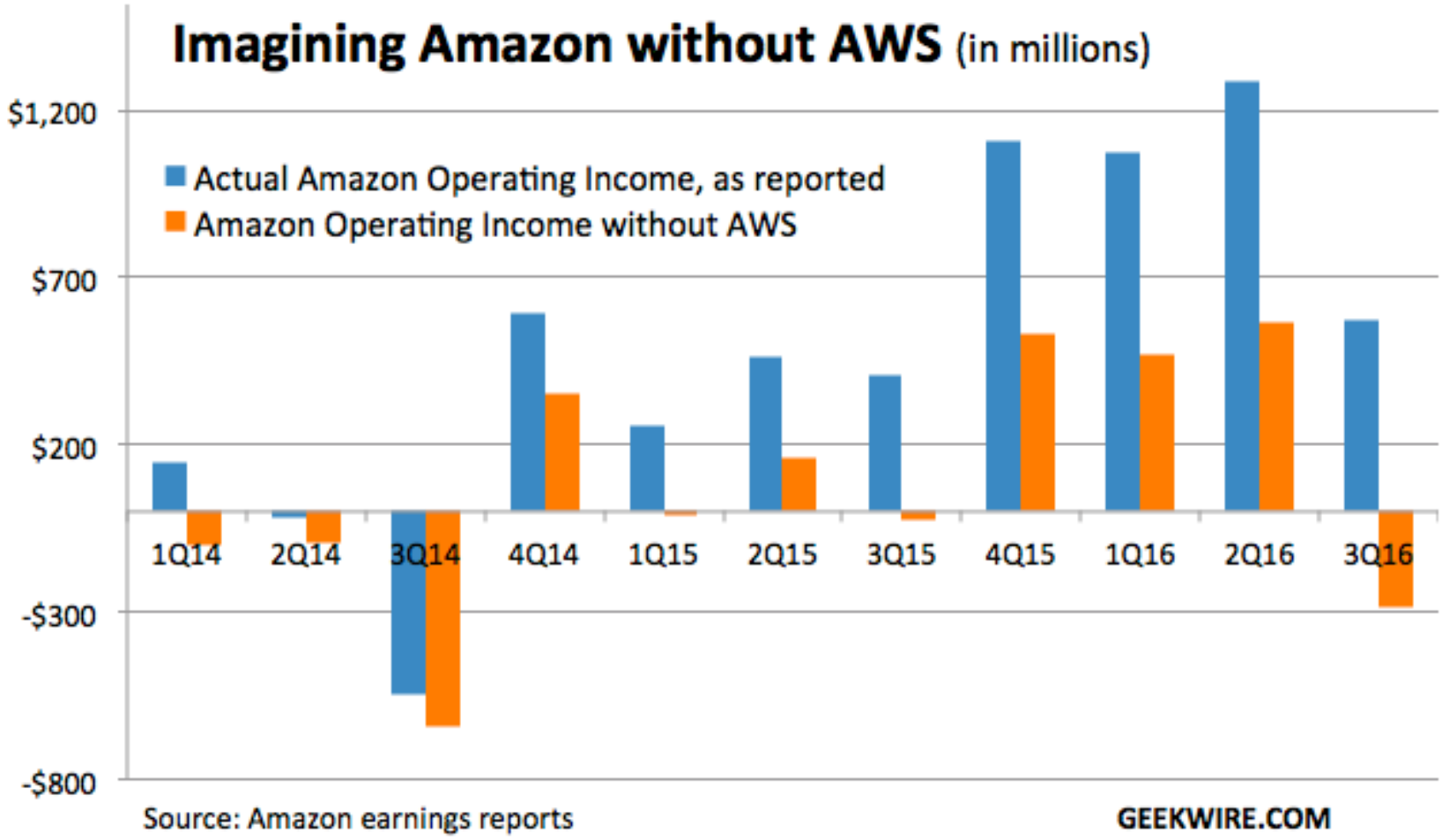
- Public cloud services spending will grow nearly seven times faster than global IT spending.
- Annual market revenue will grow at a 24.4% CAGR (compound annual growth rate) worldwide for the period 2015-2020 to total $203.4 billion.
- Forecast total $122.5 billion in 2017.
- Spending on software-as-a-service (SaaS), the predominant form of cloud computing, will lead the way. SaaS (software-as-a-service) spending will represent nearly two-thirds of global public cloud services spending this year and around 60% in 2020
- IaaS will grow at a five-year CAGR of 30.1%.
- PaaS will post a five-year CAGR of 32.2%.

Amazon


AWS year-over-year revenue growth
In percentage
80
60
40
20
69
43
43
47
49
81
78
42
1Q14
2Q14
3Q14
4Q14
1Q15
2Q15
3Q15
4Q15
1Q16
2Q16
3Q16
4Q16
1Q17
2Q17

Amazon 2Q2017



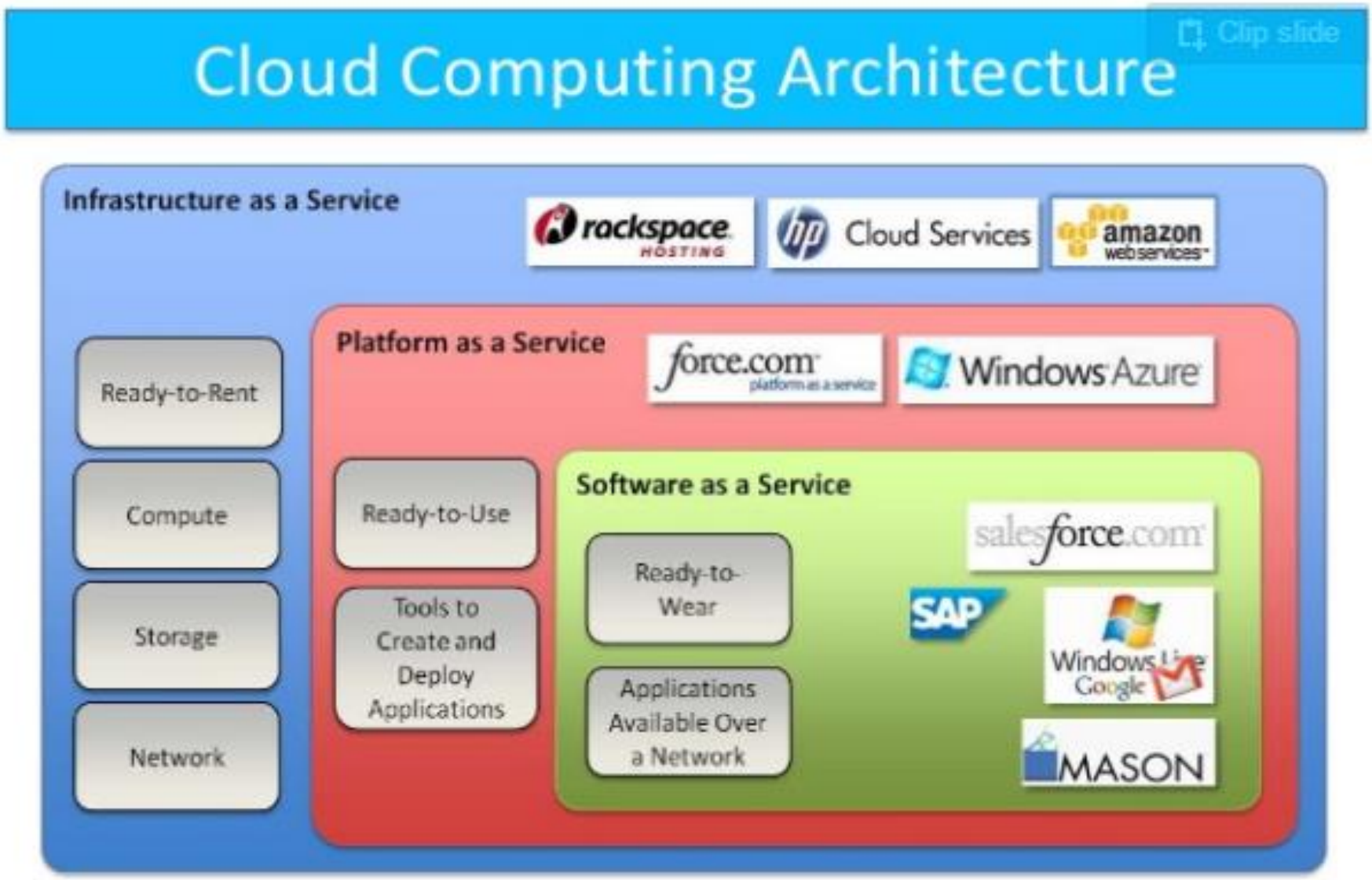

Imagining Amazon without AWS (in millions)
$1,200
$700
$200
-$300
-$800
Actual Amazon Operating Income, as reported
Amazon Operating Income without AWS
1Q14
2Q14
3Q14
4Q14
1Q15
2Q15
3Q15
4Q15
1Q16
2Q16
3Q16
Source: Amazon earnings reports
GEEKWIRE.COM

Cloud Computing

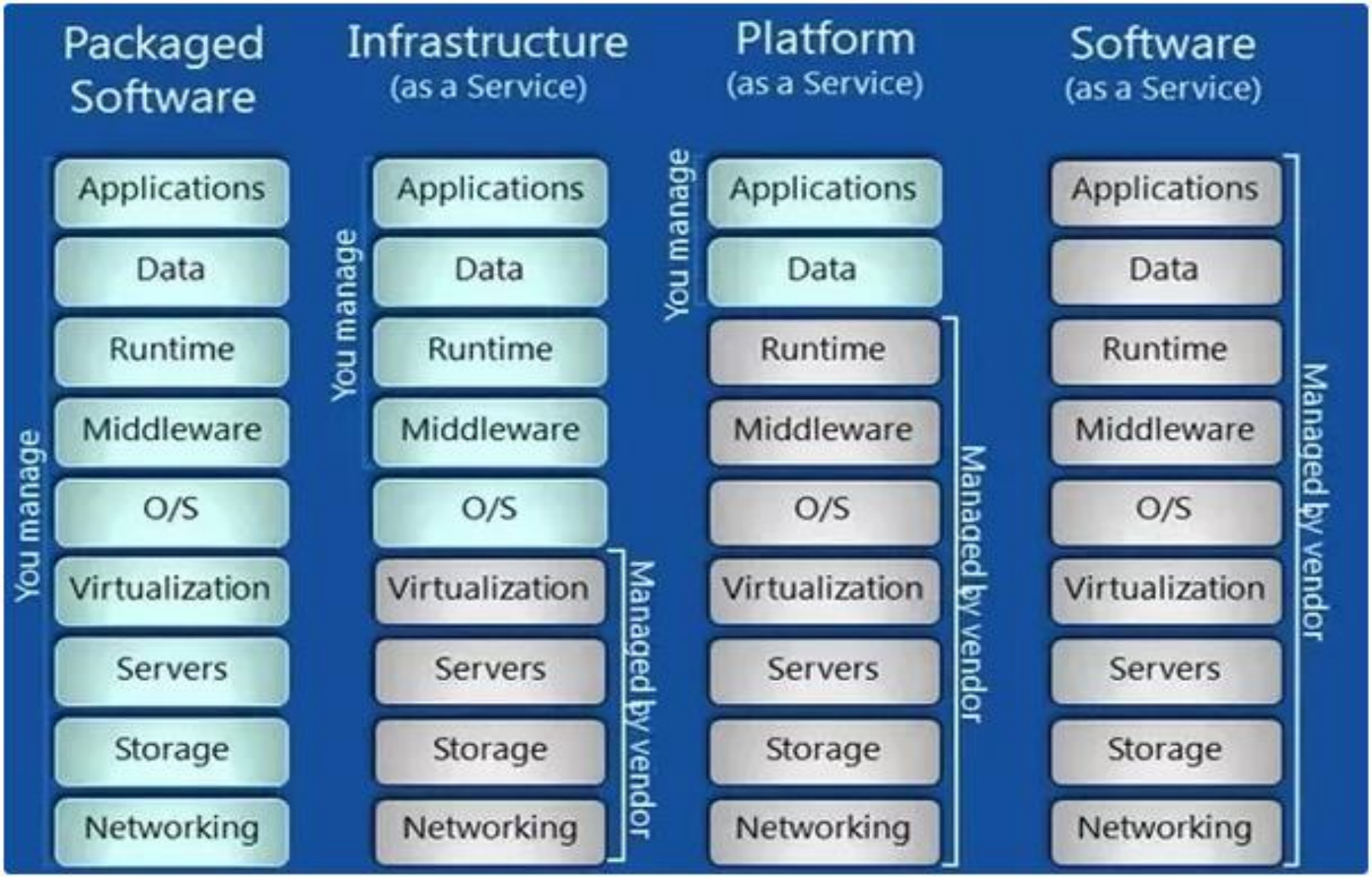
- **IaaS - Infrastructure-as-a-Service**
  - A cloud service providing infrastructure - computers, networking resources, storage. Typically virtual, but could be could be physical.

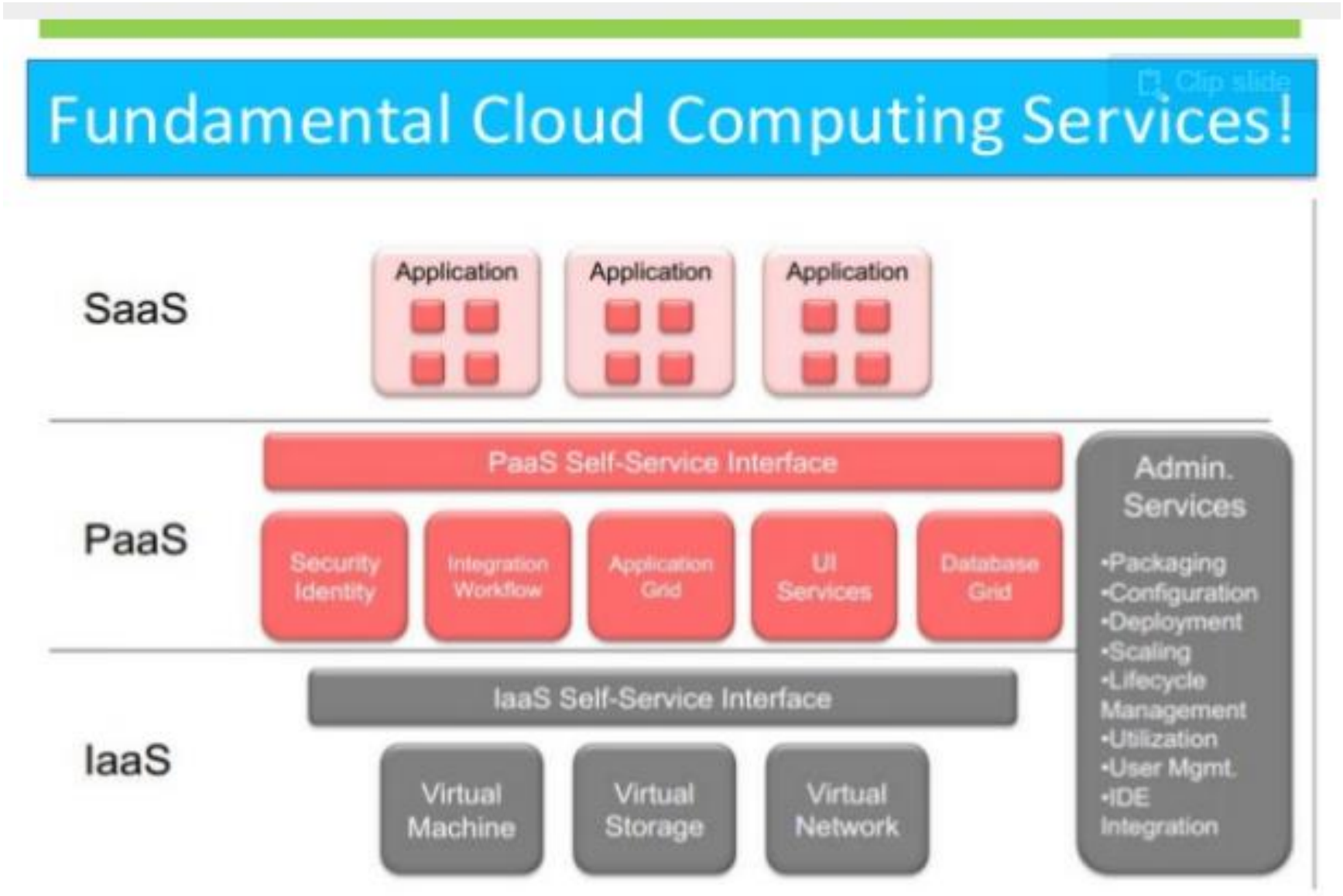
- **PaaS - Platform-as-a-Service**
  - A cloud service that hides the infrastructure (users don't see the servers, storage, switches, etc.) Provides a software development platform. Users can develop and run an application on a PaaS: the system ensures the app has the necessary infrastructure to run and scale.

- **SaaS - Software-as-a-Service**
  - A cloud service providing users access to software in a self-service, on-demand fashion. This could be a single application or an entire suite.

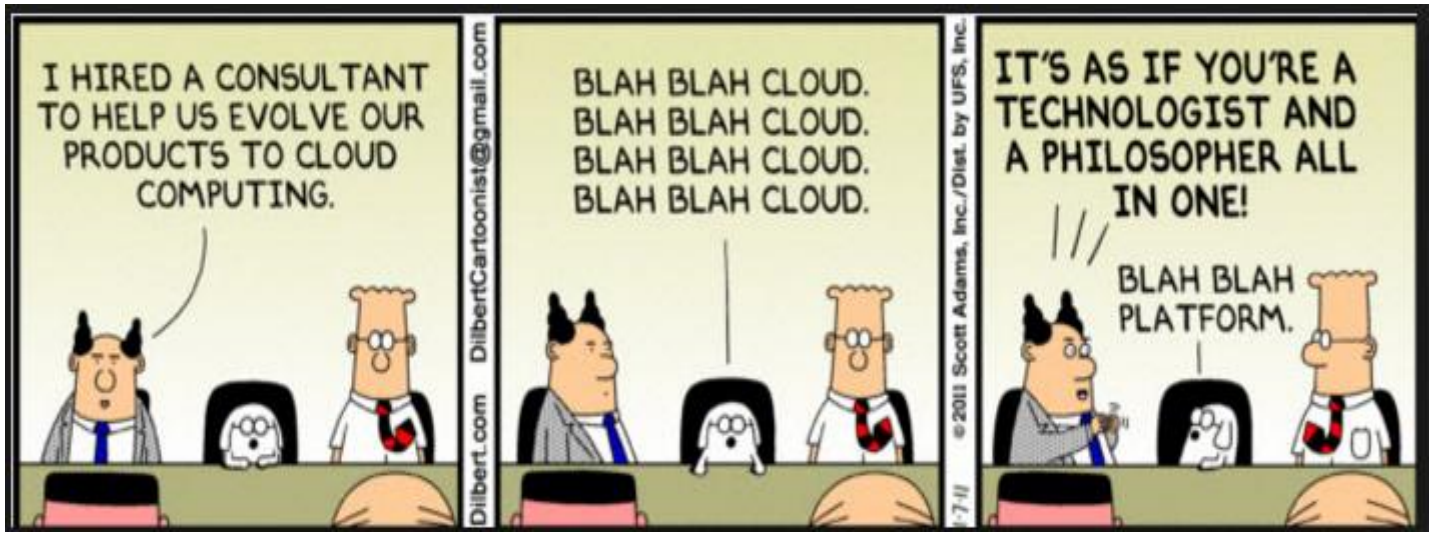

Cloud Computing Architecture
Infrastructure as a Service
rackspace HOSTING
Cloud Services
amazon web services
Ready-to-Rent
Compute
Storage
Network
Platform as a Service
force.com platform as a service
Windows Azure
Ready-to-Use
Tools to Create and Deploy Applications
Software as a Service
Ready-to-Wear
Applications Available Over a Network
salesforce.com
SAP
Windows Live
Google
MASON

Cloud Computing



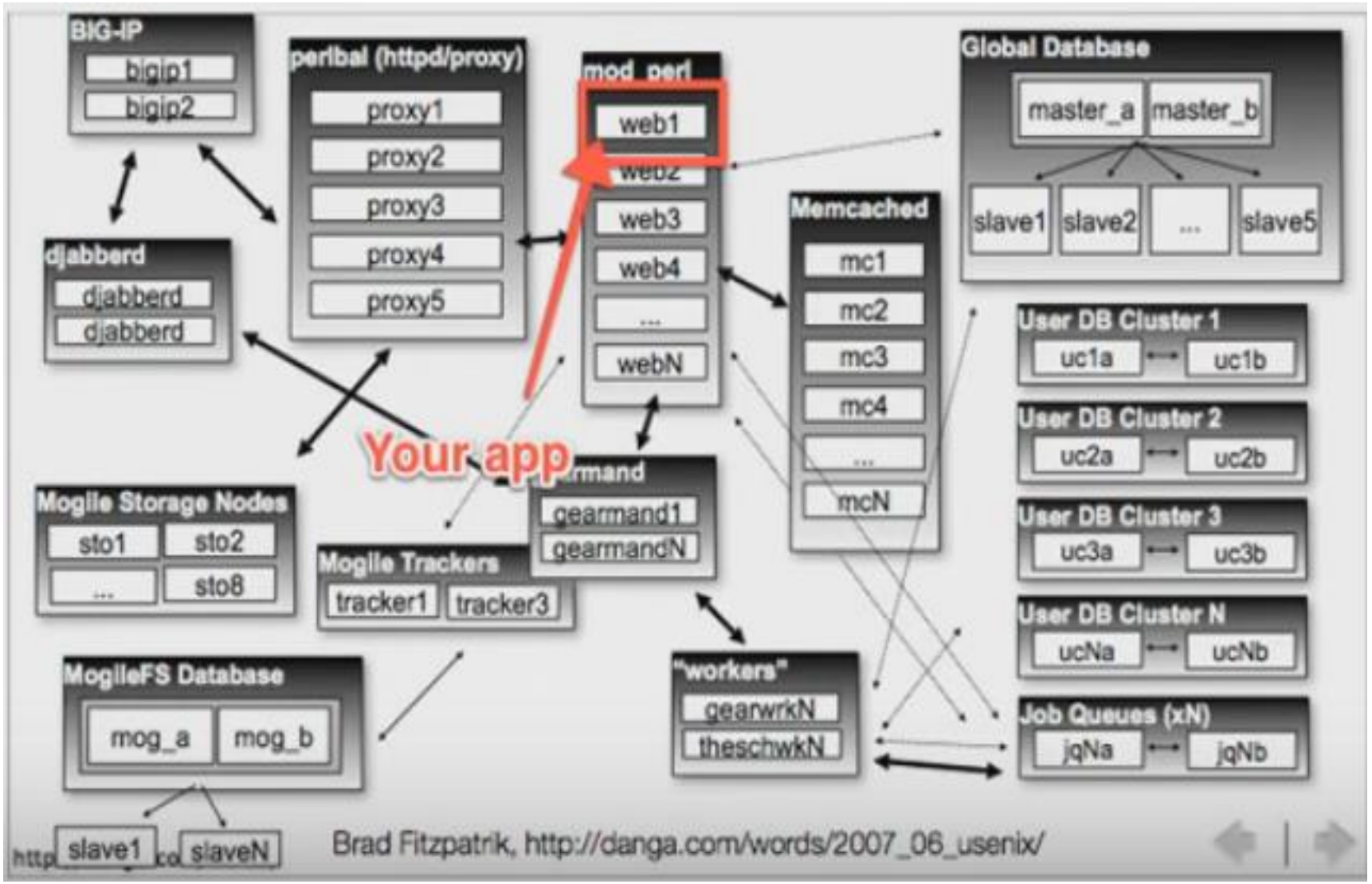

Fundamental Cloud Computing Services!
SaaS
Application
Application
Application
PaaS
PaaS Self-Service Interface
Security Identity
Integration Workflow
Application Grid
UI Services
Database Grid
Admin. Services
•Packaging
•Configuration
•Deployment
•Scaling
•Lifecycle Management
•Utilization
•User Mgmt.
•IDE Integration
IaaS
IaaS Self-Service Interface
Virtual Machine
Virtual Storage
Virtual Network

Break


I HIRED A CONSULTANT TO HELP US EVOLVE OUR PRODUCTS TO CLOUD COMPUTING.
BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD. BLAH BLAH CLOUD.
IT'S AS IF YOU'RE A TECHNOLOGIST AND A PHILOSOPHER ALL IN ONE!
BLAH BLAH PLATFORM.
Dilbert.com DilbertCartoonist@gmail.com
©2011 Scott Adams, Inc./Dist. by UFS, Inc.


I MIGRATED OUR NORTHERN DATA CENTER TO THE CLOUD.
BUT THE CLOUD STOPPED WORKING AND I CAN'T FIND THE PHONE NUMBER FOR OUR CLOUD GUY.
SO . . . WHATEVER.
YOU LOST OUR DATA CENTER?
THAT'S ONE WAY TO LOOK AT IT.
Dilbert.com DilbertCartoonist@gmail.com
7-5-13 ©2013 Scott Adams, Inc./Dist. by Universal Uclick

Heroku


BIG-IP
bigip1
bigip2
perlbal (httpd/proxy)
proxy1
proxy2
proxy3
proxy4
proxy5
mod_perl
web1
web2
web3
web4
...
webN
Global Database
master_a
master_b
slave1
slave2
...
slave5
Memcached
mc1
mc2
mc3
mc4
...
mcN
djabberd
djabberd
djabberd
Your app
User DB Cluster 1
uc1a
uc1b
User DB Cluster 2
uc2a
uc2b
User DB Cluster 3
uc3a
uc3b
User DB Cluster N
ucNa
ucNb
Mogile Storage Nodes
sto1
sto2
...
sto8
gearmand
gearmand1
gearmandN
Mogile Trackers
tracker1
tracker3
"workers"
gearwrkN
theschwkN
Job Queues (xN)
jqNa
jqNb
MogileFS Database
mog_a
mog_b
slave1
slaveN
Brad Fitzpatrik, http://danga.com/words/2007_06_usenix/

Heroku

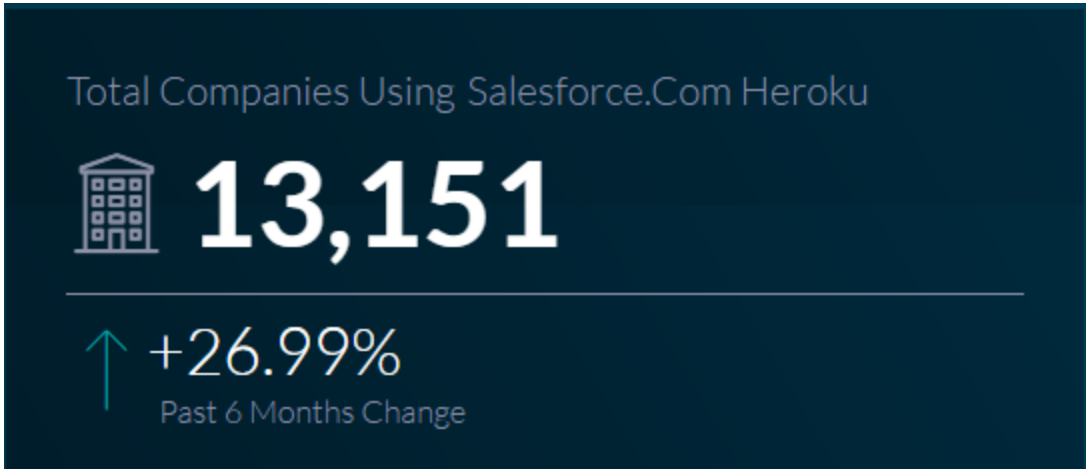
- **IaaS - Infrastructure-as-a-Service**
  - A cloud service providing infrastructure - computers, networking resources, storage. Typically virtual, but could be could be physical.

- **PaaS - Platform-as-a-Service**
  - A cloud service that hides the infrastructure (users don’t see the servers, storage, switches, etc.) Provides a software development platform. Users can develop and run an application on a PaaS: the system ensures the app has the necessary infrastructure to run and scale.

- **SaaS - Software-as-a-Service**
  - **A cloud service providing users access to software in a self-service, on-demand fashion. This could be a single application or an entire suite.**

Heroku

- **Heroku**
  - One of the first PaaS providers out there (June 2007)
  - Initially focused on Ruby
  - Purchased by SalesForce in December 2010
  - Free on a small scale (great for students)
  - Many competitors
    - AWS, GoogleSites, Azure, OpenShift, DigitalOcean



Heroku


G2 Crowd Grid℠ for PaaS
Summer 2015
Contenders
Leaders
Microsoft Azure
Heroku
Google App Engine
AWS
Elastic Beanstalk
OpenShift
Cloud Foundry
DigitalOcean
Niche
High Performers
Market Presence
Satisfaction
Share Image

Heroku


Everything you need to build, run, and scale customer apps
Dynos
Run virtually any language at scale
Database
Enterprise grade PostgreSQL as a Service
Add-ons
Marketplace for data stores and app services

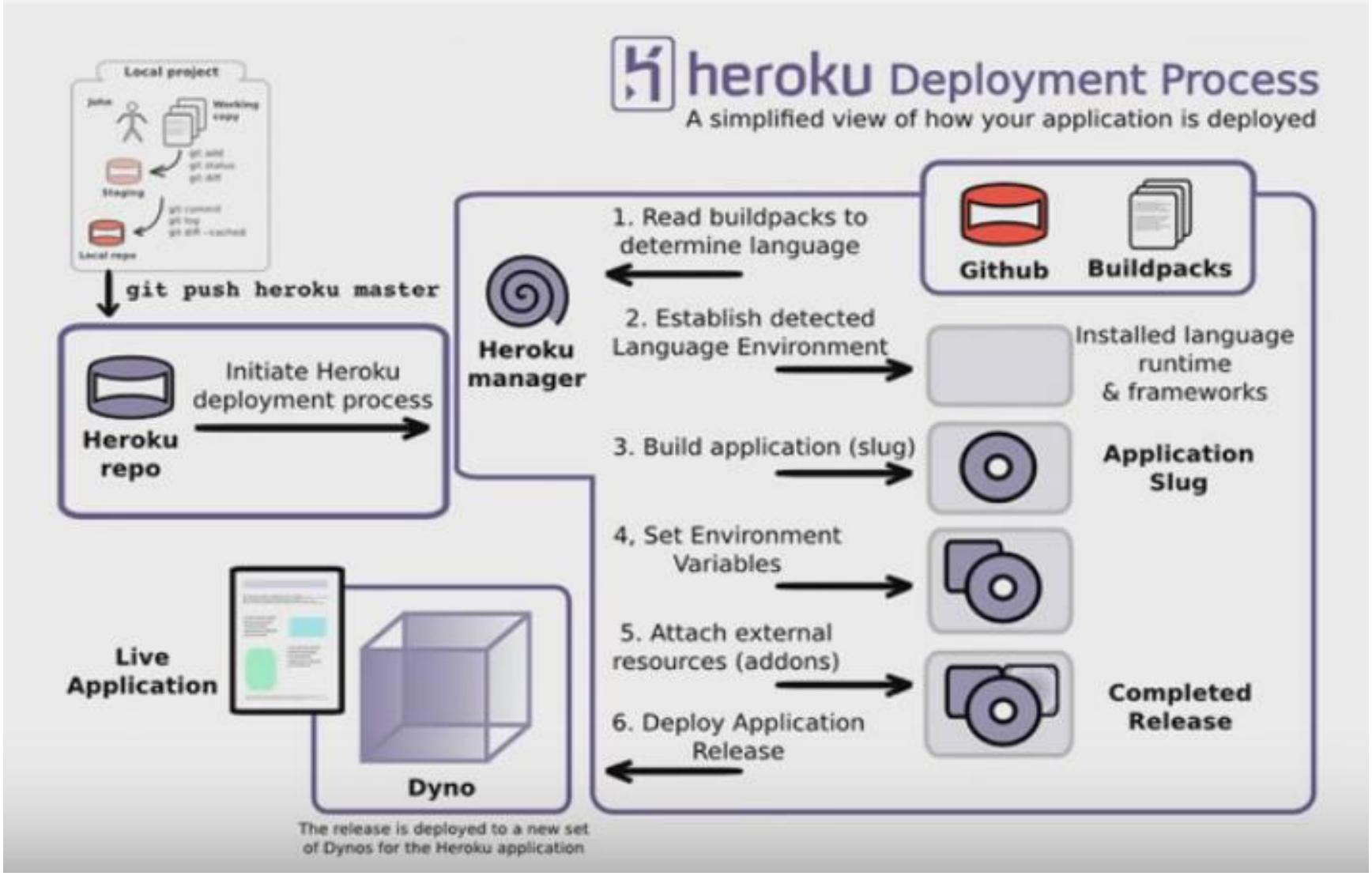
Heroku

- **Provides users with one or more DYNOs**
  - “Dyno” - a virtual engine that runs your app.
  - If you need to add capacity, you can add more dynos.
- **Provides Standard PostgreSQL database engine**
- **Allows “add-on’s”**
  - Load testing
  - Email sending
  - Create a PDF
  - Alternate DB (Mongo or Redis)
- **Core support for Ruby, Python, NodeJS, Java, PhP**
- **Tightly integrated with git for version control**

Heroku

- **Write your app in any supported language**
- **Ready to deploy, push to heroku git master**
- **Heroku Manager**
  - Packages your app into an executable bundle
  - Contains all components needed to run your app
  - Compiled app is called a “slug”
  - Slug is executable on the DYNO
  - Compiled through a “buildpack”
  - Every change is a “new release” – easy to roll back

Heroku


heroku Deployment Process
A simplified view of how your application is deployed
Local project
git push heroku master
Heroku repo
Initiate Heroku deployment process
Heroku manager
1. Read buildpacks to determine language
Github
Buildpacks
2. Establish detected Language Environment
Installed language runtime & frameworks
3. Build application (slug)
Application Slug
4, Set Environment Variables
5. Attach external resources (addons)
6. Deploy Application Release
Completed Release
Live Application
Dyno
The release is deployed to a new set of Dynos for the Heroku application

Heroku

- **Learn More At**

https://devcenter.heroku.com/categories/reference

Heroku

- **Demo** (first delete my old app!)
  - Running Windows CMD console
  - Make sure I've got git on my PC
  - Download and install Composer on my PC (buildpack for php)
  - Download and install Heroku ToolBelt on my PC
  - Check versions of git, heroku, composer "`composer --version`"
  - "`git clone https://github.com/heroku/php-getting-started.git`"
  - Create a git repo on my PC (`git init`)
  - Edit my app (a php program)
  - Log in to Heroku
  - Do a "`heroku create`" to create the app in Heroku
    - It gives you a URL
    - Go look at it
  - Add my file to stage, then commit my app in git
  - Do a "`git push heroku master`"
  - Look at the app – change it again and push
  - Look at heroku logs "`heroku logs --tail`"

Heroku

- **Demo**
  - Running Windows CMD console
  - Make sure I've got git on my PC
  - Download and install Composer on my PC (buildpack for php)
  - Download and install Heroku ToolBelt on my PC
  - Check versions of git, heroku, composer "`composer --version`"
  - "`git clone https://github.com/heroku/php-getting-started.git`"
  - Create a git repo on my PC (`git init`)
  - Edit my app (a php program)
  - Log in to Heroku
  - Do a "`heroku create`" to create the app in Heroku
    - It gives you a URL
    - Go look at it
  - Add, then commit my app in git
  - Do a git push heroku master
  - Look at the app
  - Look at heroku logs "`heroku logs --tail`"