greatlearning

PGP in Cloud Computing

Course | Cloud Foundations

Nirmallya Mukherjee

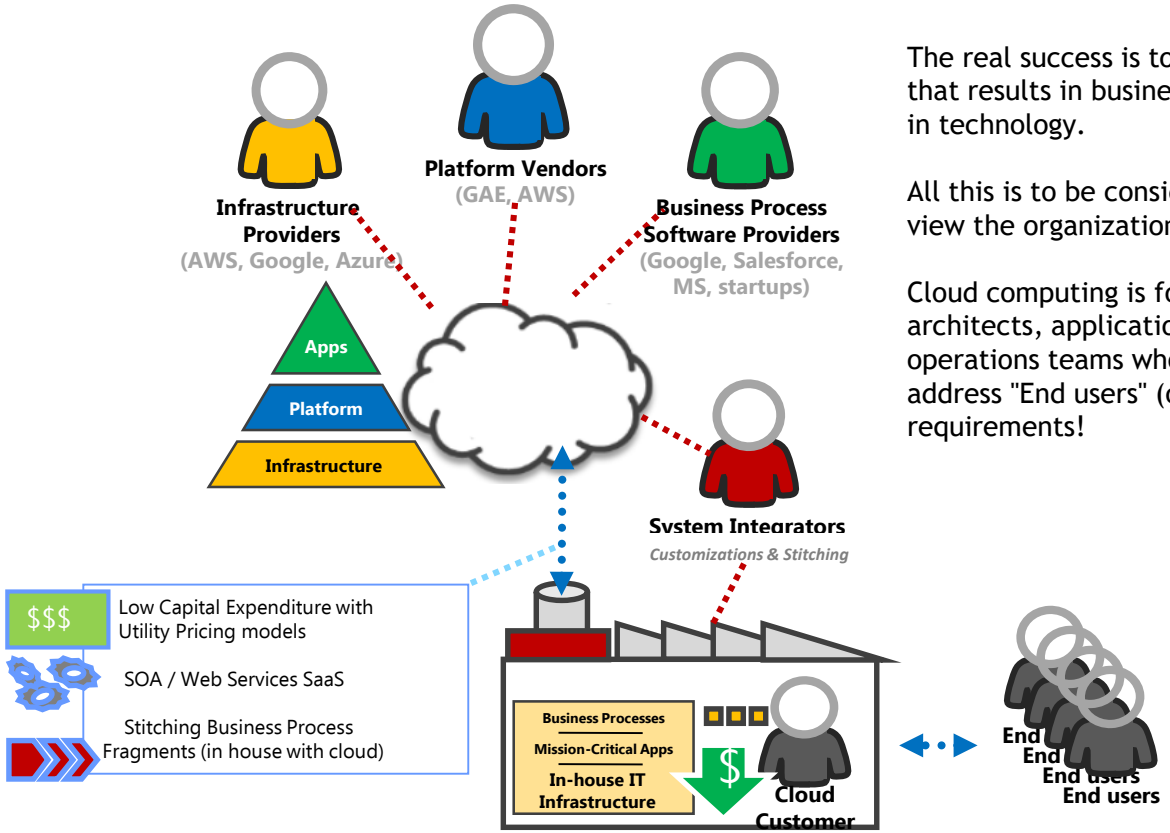
Ex Chief Architect – Dell | Cloud Expert | Developer

Apart from my laptop (I5, 4cpu, 8GB, 1TB, Ubuntu), I have other machines but I let others (AWS & Google) host it for me ...

There is nothing called as "Cloud computing"

It's someone else's computer in their datacenter!

Our focus



Not all applications can be or should be moved to the cloud.

The real success is to have the right blend that results in business benefits and ease in technology.

All this is to be considered keeping in view the organization's growth roadmap.

Cloud computing is for business leaders, architects, application designers & operations teams who are looking to address "End users" (or organizations) biz requirements!

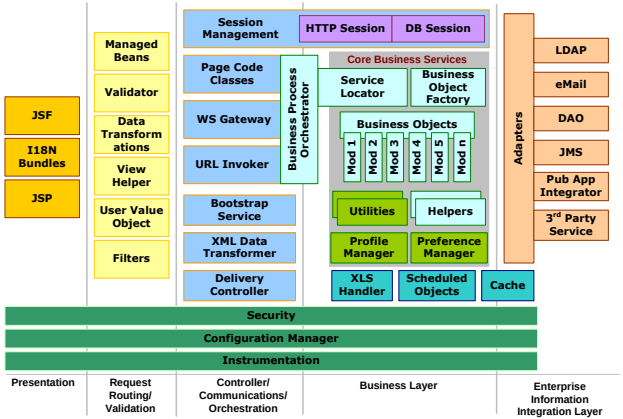
A few stories

- Instagram launch
- Flipkart billion day sale
- Did you hear about ASP? (this was early 2000)
- How many of us want to build or have already installed a nuclear power plant at home?
- One final thing ... familiar with SETI@HOME?

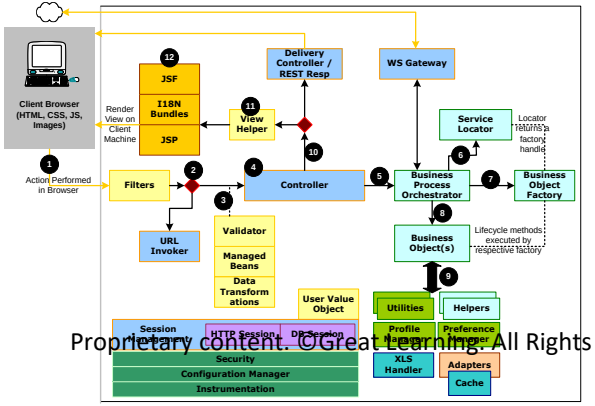
What does the business worry about?

- Service availability
- Data lock-in
- Redundancy
- Confidentiality
- Compliance
- Audit trail
- Unpredictable user growth
- Infrastructure shortage & refresh
- Provisionig latency
- Multi datacenter management
- Elasticity
- Licencing fee

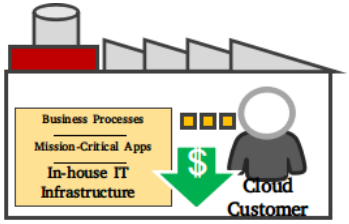
The classical enterprise



Monolith Web application - all components are bundled together as a single deployable/application



1. Portals
2. Search
3. Content management
4. Middleware/ESB
5. BPM
6. Database farm
7. Warehousing
8. ETL processes
9. BI/Reporting
10. CSR product
11. ERP product
12. Contact center product
13. Infra monitoring tools
14. Code repo/CI/CD tools
15. Productivity/Office/Collaboration tools
16. Operating systems
17. Virtualization software
18. Infrastructure h/w (machines, routers)
19. Networking (h/w firewalls, load balancers)
20. Buildings & perimeter security
21. Electricity (primary, secondary)/Cooling/Land



+

1. Human expertise & capital
2. Ongoing process of patches & upgrades
3. Procurement department, many 3rd party vendors
4. Dissonant operations/ownership
5. Utilization challanges & capacity guesswork
6. Various licensing (fixed + incremental) & AMC
7. Capital expenses & depreciating assets
8. Disconnected ops expenses from top line
9. Some intangibles like "stress" too!

Why cloud?

greatlearning

Python/Java/C++ looks & runs nice on the cloud

OR

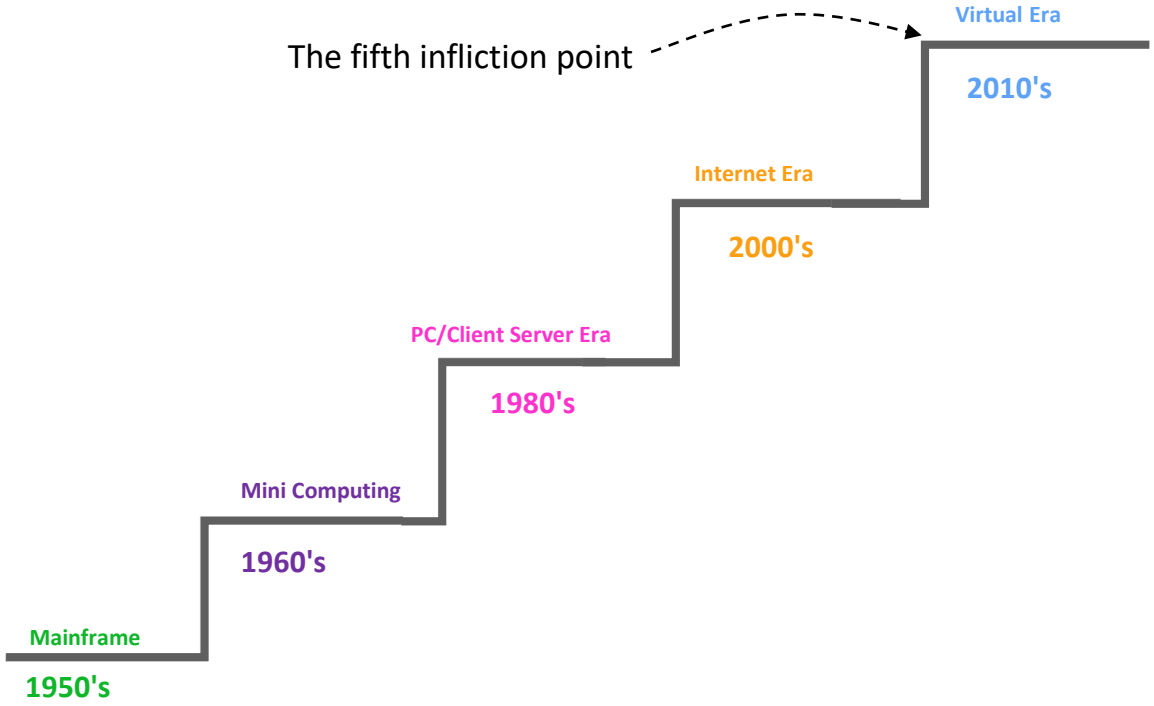
$$\text{UserHours}_{\text{cloud}} \times (\text{Revenue} - \text{Cost}_{\text{cloud}}) \geq \text{UserHours}_{\text{datacenter}} \times [\text{Revenue} - (\text{Cost}_{\text{datacenter}} / \text{Utilization})]$$

The left-hand side multiplies the net revenue per user-hour by the number of user-hours, giving the expected profit from using Cloud Computing.

The right-hand side performs the same calculation for a fixed-capacity datacenter by factoring in the average utilization, including nonpeak workloads, of the datacenter.

Whichever side is greater represents the opportunity for higher profit.

A short history and evolution



Any definitions?

greatlearning

- Style of computing in which **massively scalable** IT related capabilities are provided "**as a service**" using internet technologies to multiple **"external customers" –** *Gartner*

- Pool of abstracted, **highly scalable**, and managed compute infrastructure capable of hosting end-customer applications and **billed by consumption –** *Forrester*

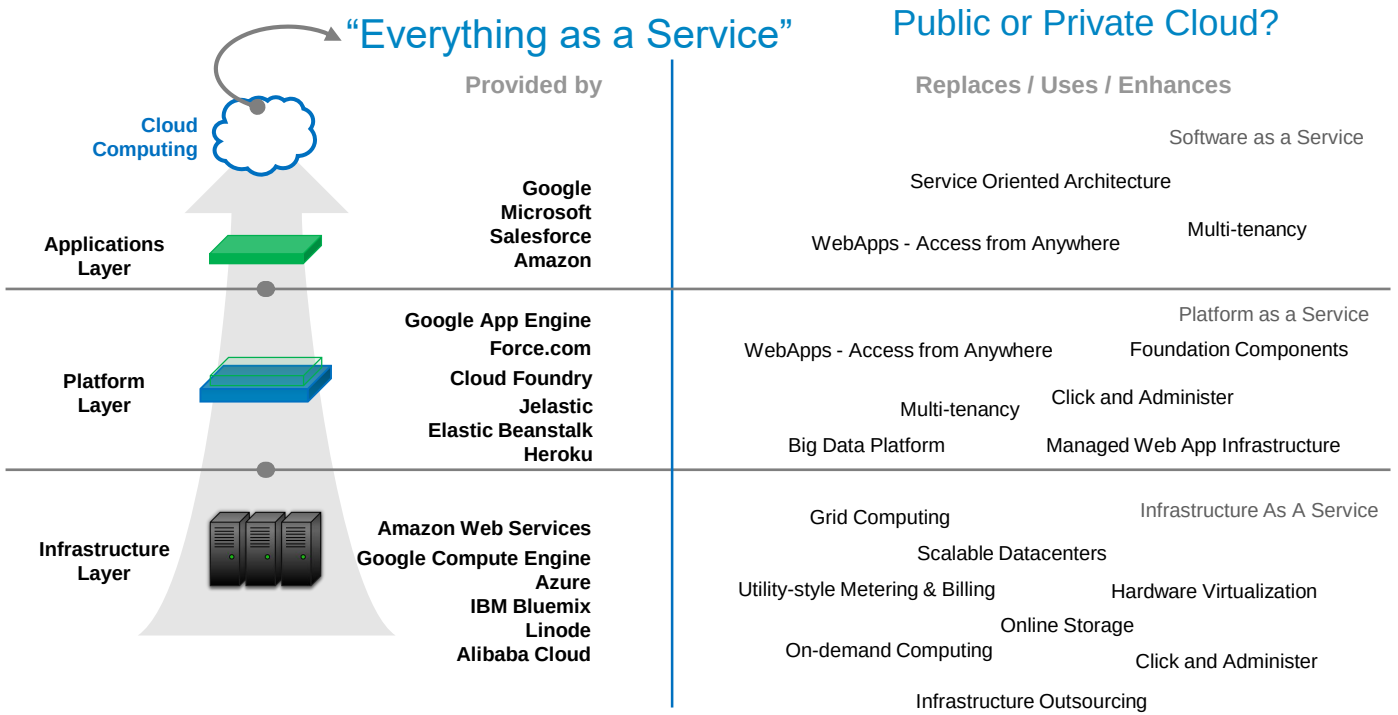
Myths of cloud computing

greatlearning

- There’s one single “Cloud”
- All you need is your credit card
- The cloud always saves you money
- The cloud always reduces your workload
- Integration (two versions)
  - You can seamlessly blend your private “Cloud” (your virtualized datacenters) with public providers
  - You won’t ever be able to seamlessly blend your public and private clouds
- A cloud provider can guarantee security
- If you are using virtualization, you are doing cloud computing
- Cloud computing is only about technology

Service delivery models

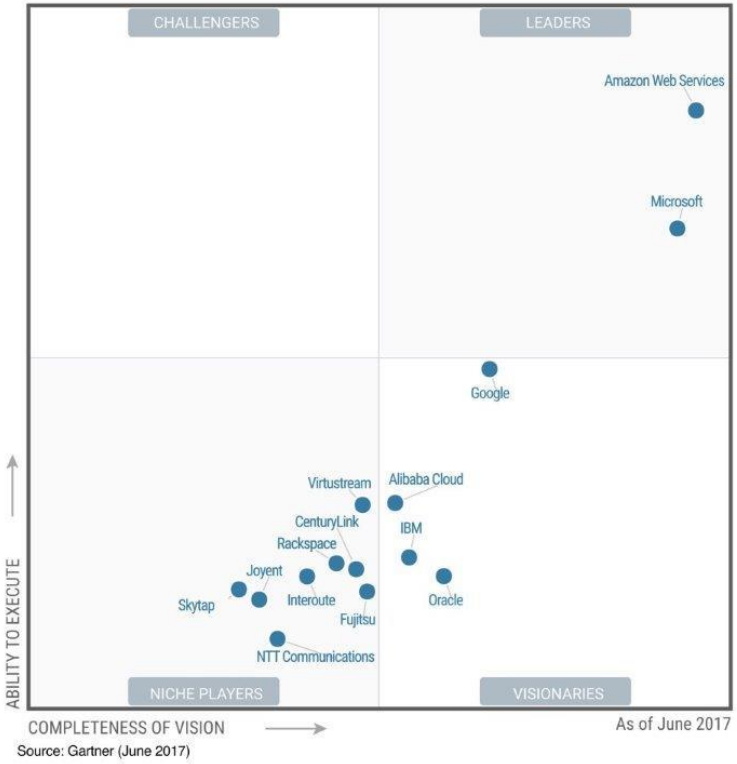
Gartner: a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies.



Cloud providers comparison

greatlearning

Figure 1. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide



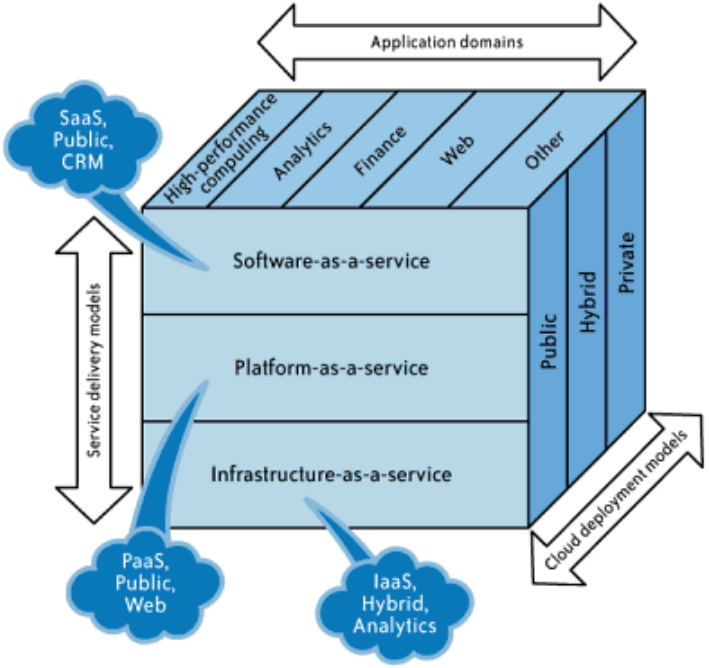
:

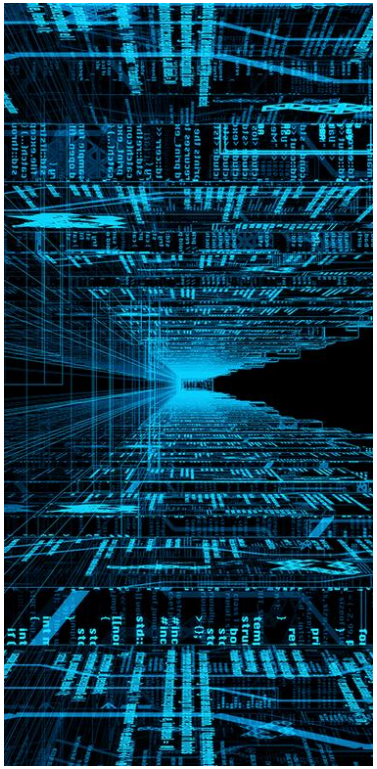
http://www.infoworld.com/article/3150205/cloud-computing/cloud-compute-aws-azure-google-softlayer-compared.html

http://www.infoworld.com/article/3132023/security/10-aws-security-blunders-and-how-to-avoid-them.html#tk.drr_mlt

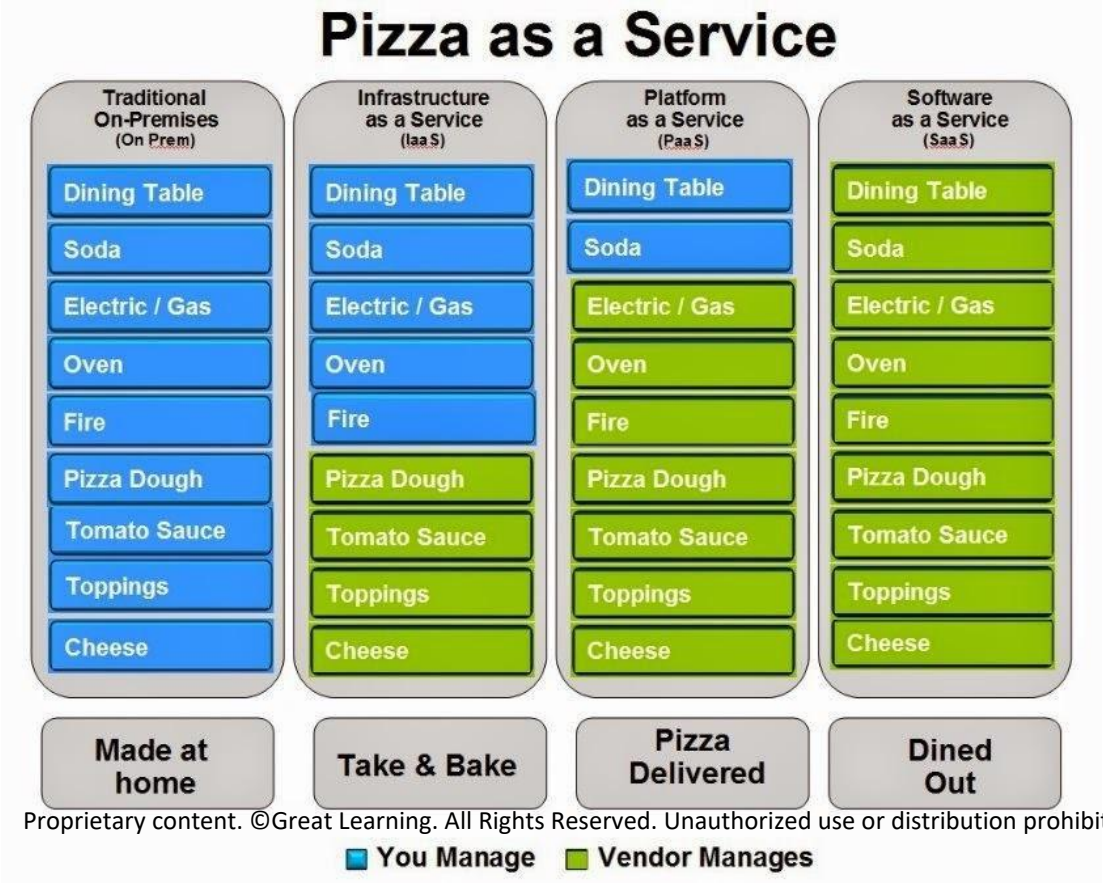
SPIDERS

"**S**aaS", "**P**aaS", "**I**aaS", "big**D**ata", "**E**lastic", "**R**esilient" & "**S**ubscription"

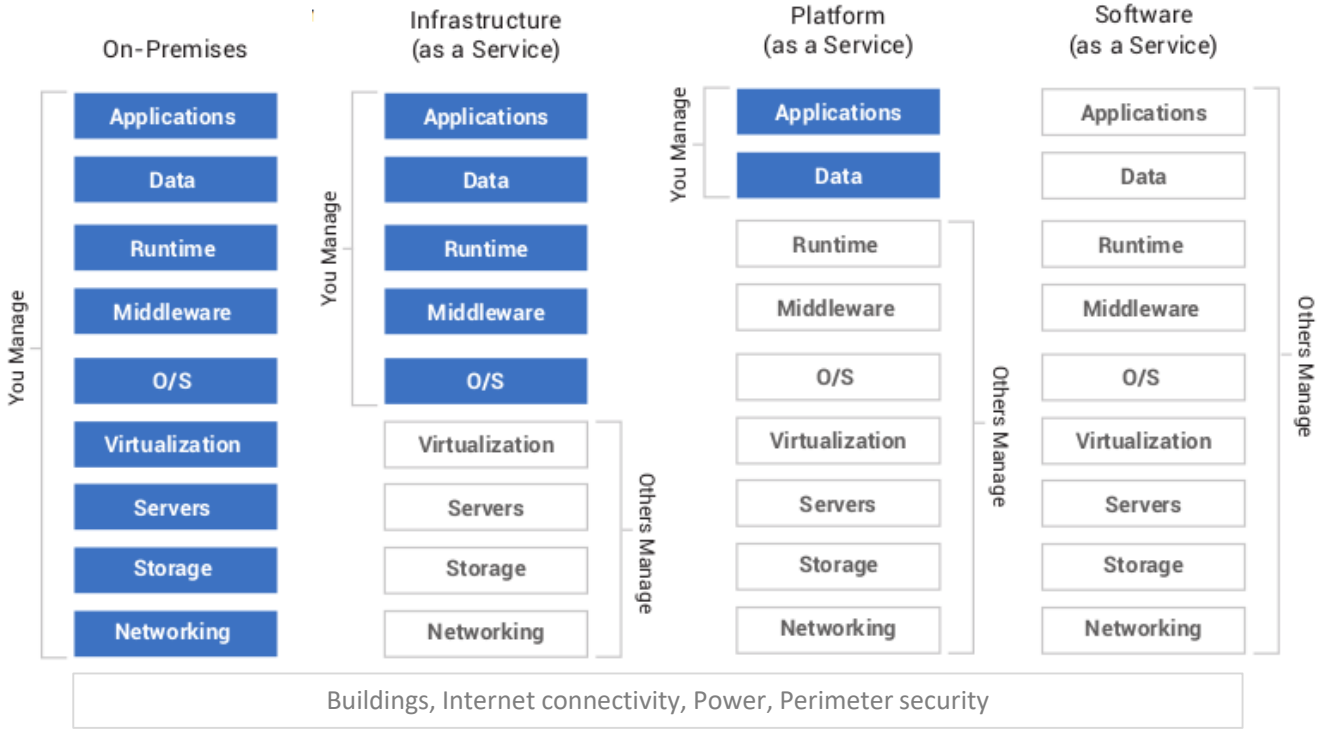




A perspective

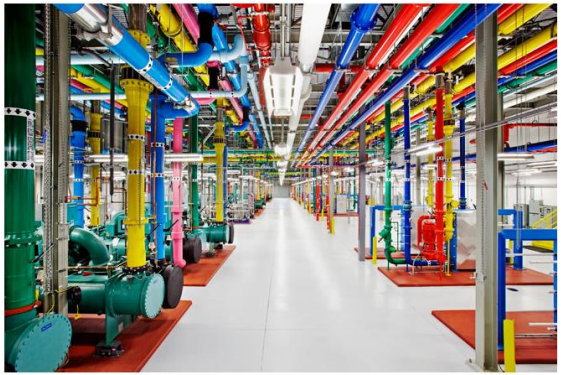


Degree of abstraction - app view



Where are these things coming from?

A server room in Council Bluffs, Iowa.
Photo: Google/Connie Zhou



A central cooling plant in Google's Douglas County, Georgia, data center.
Photo: Google/Connie Zhou

Let's see it

https://www.youtube.com/watch?v=avP5d16wEp0

Cloud computing attributes

greatlearning

Choice of provider - Based on business need and partnership models. Market competition is good for consumers as it avoids monopoly

Agility, OnDemand, Self service - Procurement latency is not longer a barrier

Resource pooling - leverage economies of scale for cost reduction

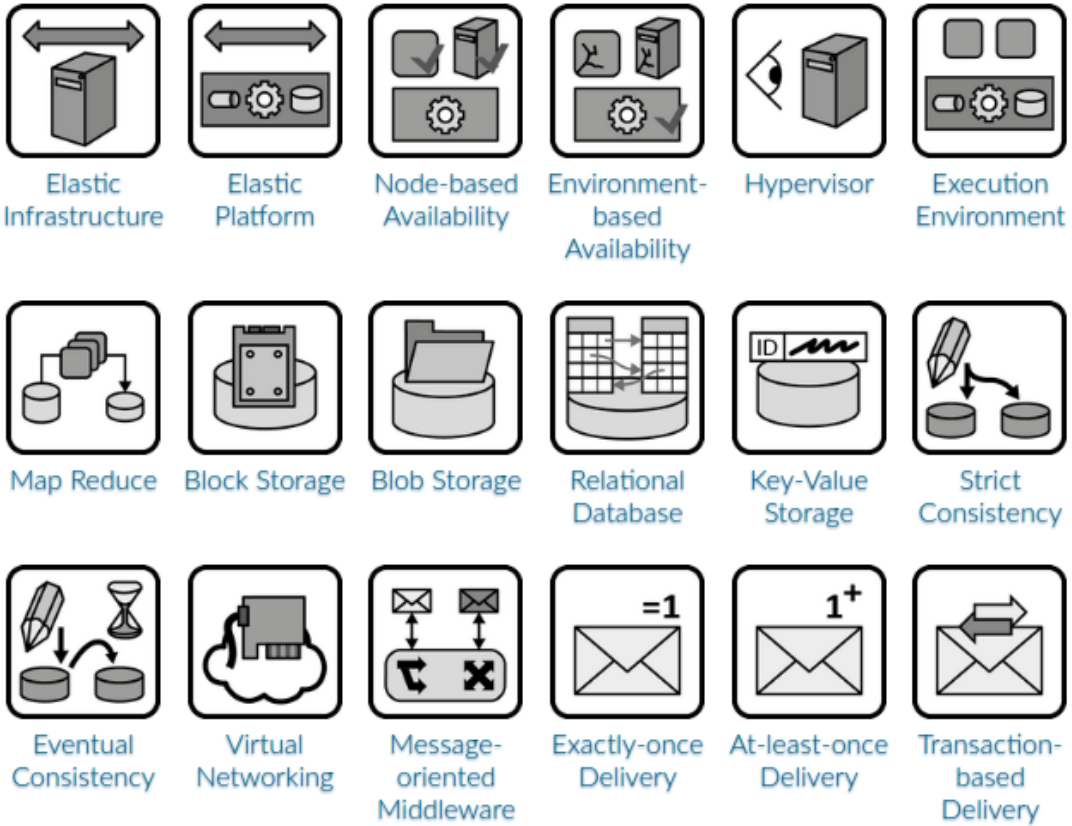
Admin & Monitoring - being proactive is better than reactive

Resilient, Elastic & Subscription - build architecture to sustain failures, no need to plan infrastructure ahead of time based on speculation & convert capex to opex

Core data center services - Compute, Storage & Networking, focus more on business rather than data center operations

Managed services - hosted managed services allow developers to focus on core applications with business logic, support services become an API call

Cloud offerings



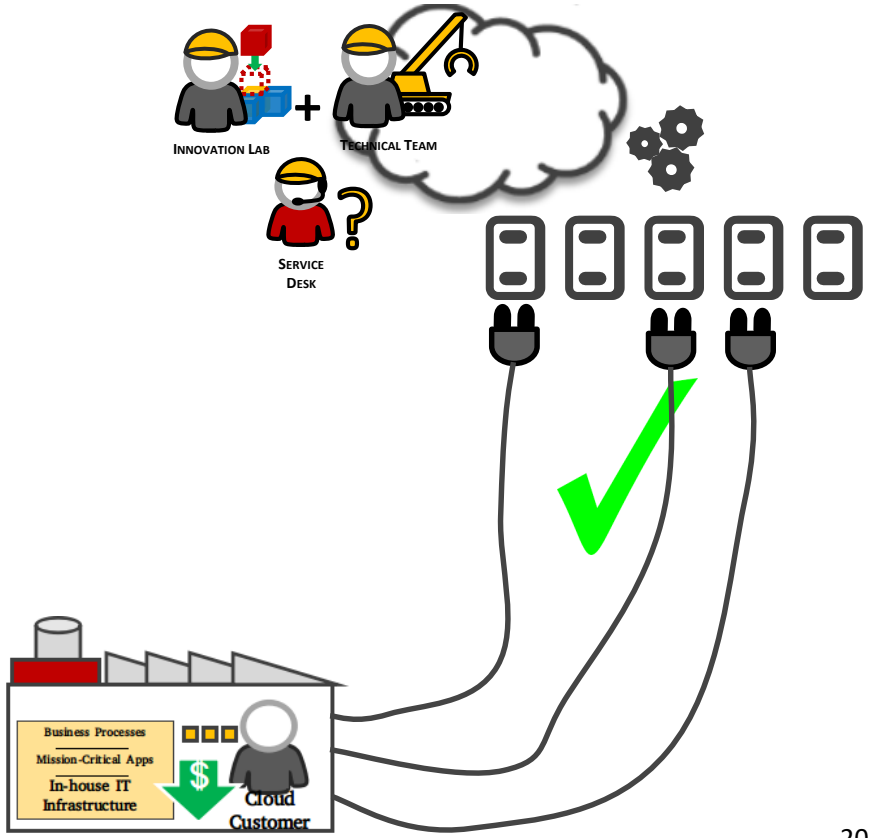
Hosted managed services

greatlearning

1. Portals
2. Search
3. Content management
4. Middleware/ESB
5. BPM
6. Database farm
7. Warehousing
8. ETL processes
9. BI/Reporting
10. CSR product
11. ERP product
12. Contact center product
13. Infra monitoring tools
14. Code repo/CI/CD tools
15. Productivity/Office/Collaboration tools
16. Operating systems
17. Virtualization software
18. Infrastructure h/w (machines, routers)
19. Networking (h/w firewalls, load balancers)
20. Buildings & perimeter security
21. Electricity (primary, secondary)/Cooling/Land

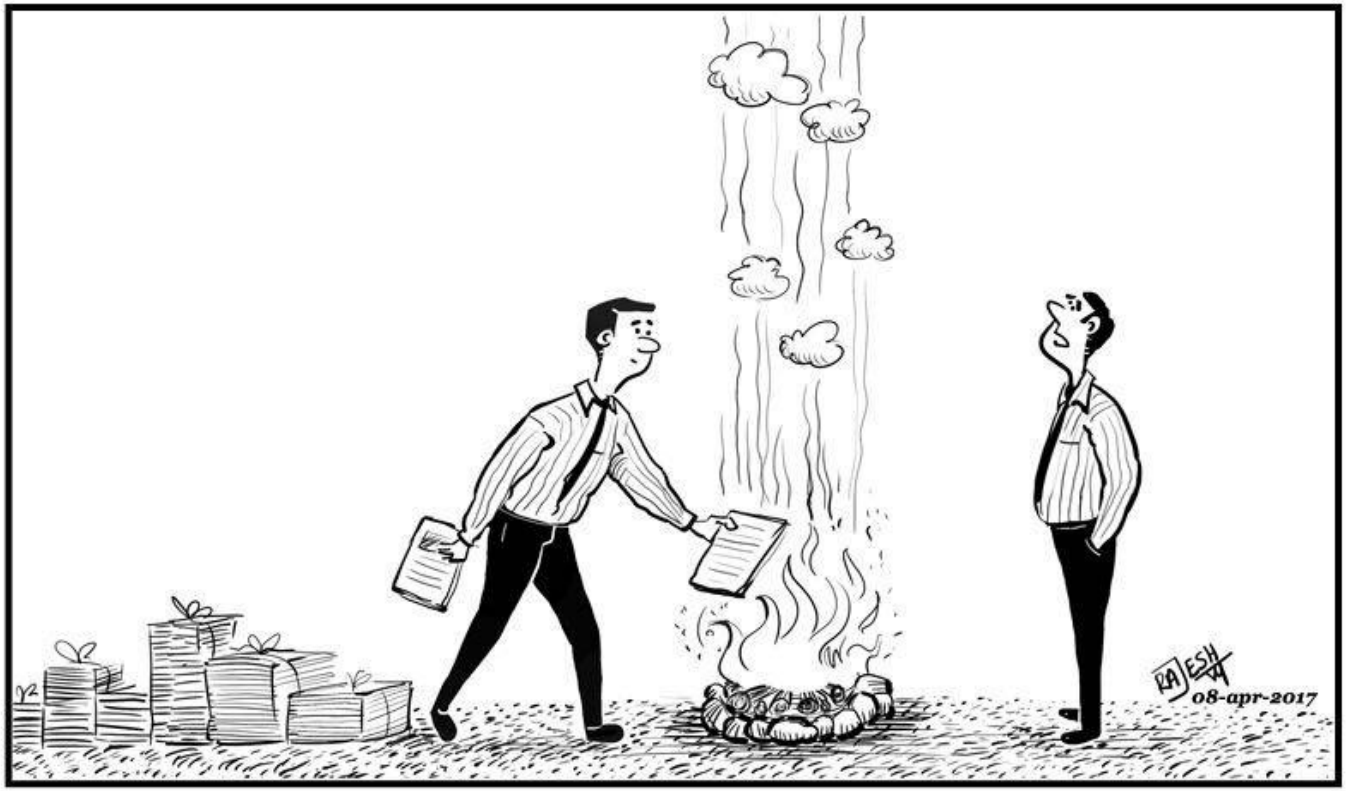
+

1. Human expertise & capital
2. Ongoing process of patches & upgrades
3. Procurement department, many 3rd party vendors
4. Dissonant operations/ownership
5. Utilization challanges & capacity guesswork
6. Various licensing (fixed + incremental) & AMC
7. Capital expenses & depreciating assets
8. Disconnected ops expenses from top line
9. Some intangibles like "stress" too!



Cloud storage as a service

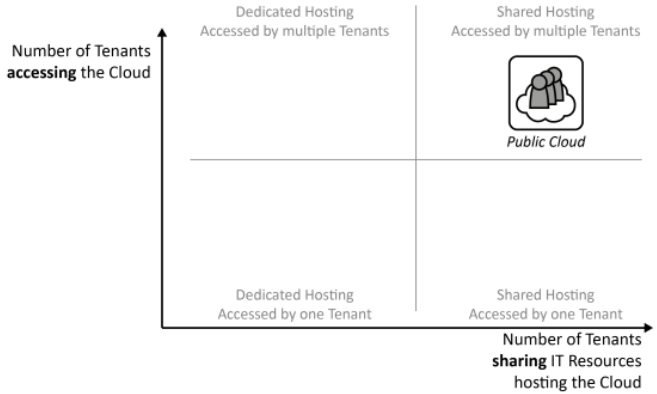
greatlearning



"Are you sure this is how we upload data into the Cloud? "

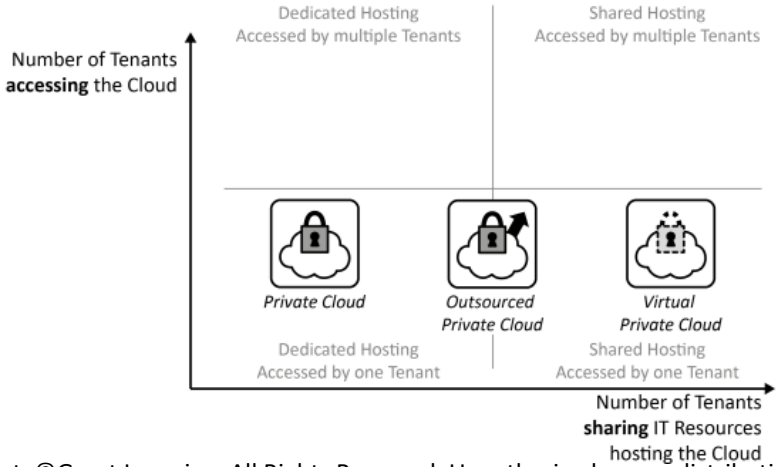
Cloud Deployment Models

- Public
  - A model where a service provider makes resources, such as applications and storage, available to the general public over the internet
  - The hosting environment is shared between many customers possibly reducing the costs for an individual customer
  - Leveraging economies of scale enables a dynamic use of resources, because workload peaks of some customers occur during times of low workload of other customers
  - It is hosted and managed by a 3rd party from one or more data centers
  - What does this mean about a customer's data?



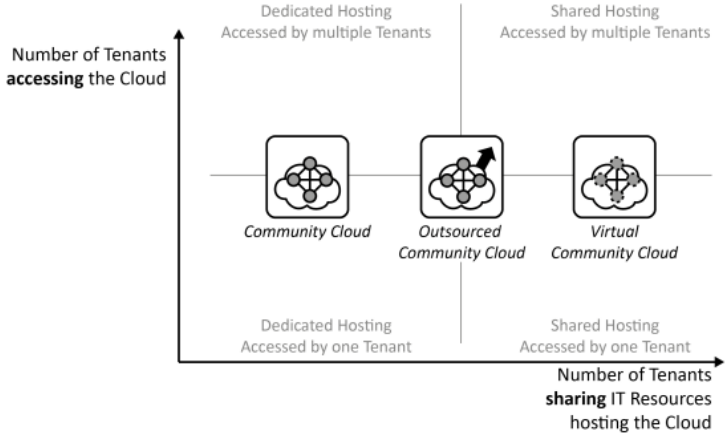
Cloud Deployment Models

- Private
  - Cloud computing properties are enabled in a company-internal data center
  - Alternatively, the Private Cloud may be hosted exclusively in the data center of an external provider, then referred to as outsourced Private Cloud
  - Public Cloud providers also offer means to create an isolated portion of their cloud made accessible to only one customer: a Virtual Private Cloud (aka Private Networking) which is the default behavior for many public cloud providers



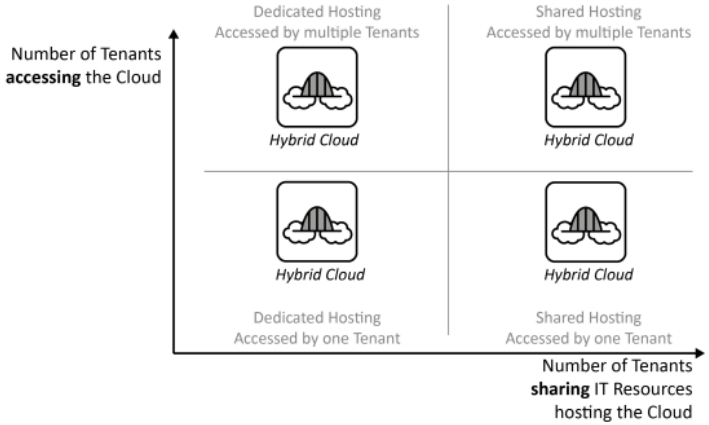
Cloud Deployment Models

- Community
  - IT resources required by all collaborating partners are offered in a controlled environment accessible only by the community of companies that generally trust each other
  - Carving out a dedicated area exclusively for a company could be possible (private)
  - A similar model to that of ASP, completely managed by a 3rd party (outsourced)

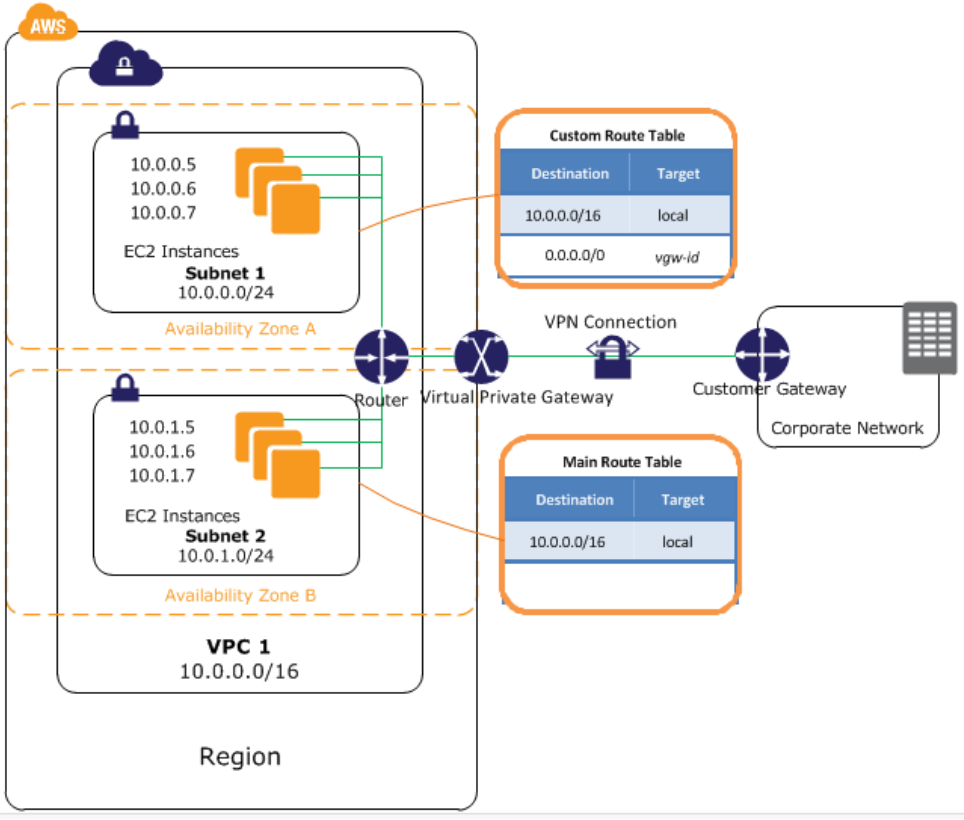


Cloud Deployment Models

- Hybrid
  - Any combination of Public, Private & Community
  - Eg any cloud (one or more) along with any static in-house data center (one or more) are integrated
  - Applications can chose the right environment leveraging the best from each option
  - Enabled in "Cloud bursting"
  - Interconnecting usually happens via VPN



An Amazon example



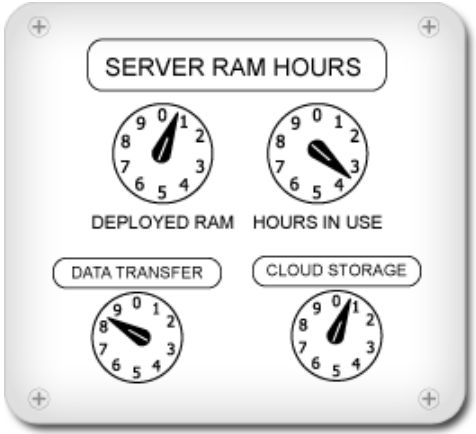
Debate

Analyze these scenarios -

- We pay to the local electricity provider at the end of the month
- We pay the grocery store after shopping
- Our phone bill varies each month
- The vehicle needs fuel depending on the distance it runs
- What are these examples of?

Subscription model

- Service subscription
  - Customers avoid large upfront capital expense
  - Pay as an ongoing operational expense
  - Easily and quickly scale up or down based on business demand and only pay for what is needed (Economies of scale)
  - Better matches today’s financial drivers
  - No "buying", its always "renting"

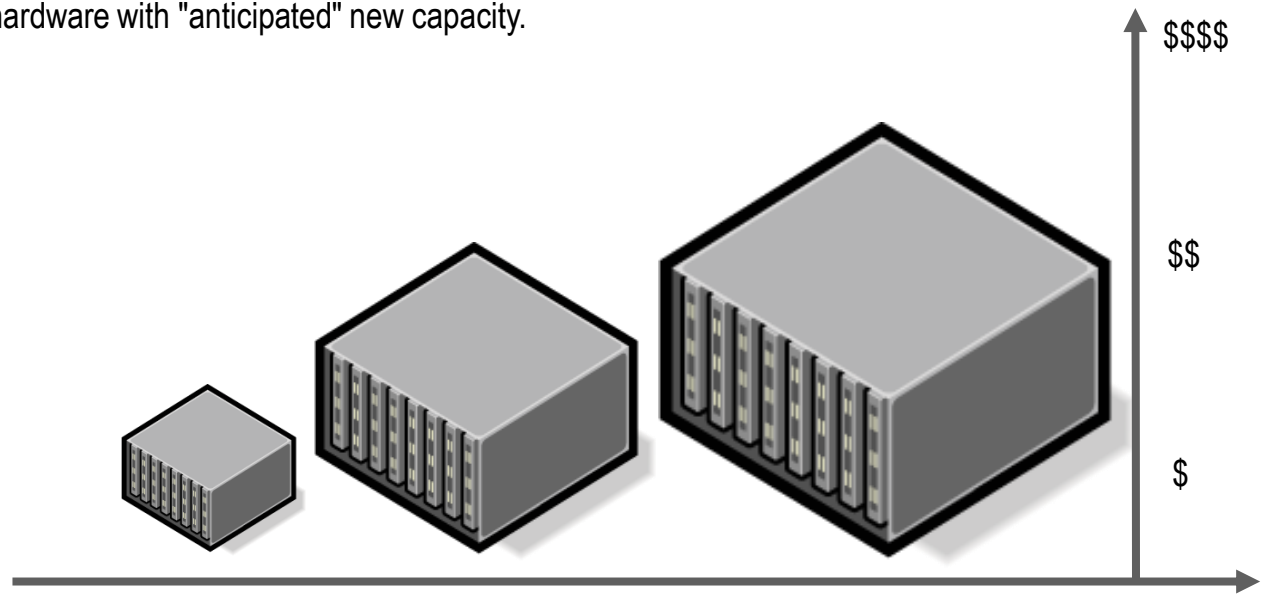


Classical Scaling model

greatlearning

Previous generation of application architectures required larger and larger servers to handle capacity needs.

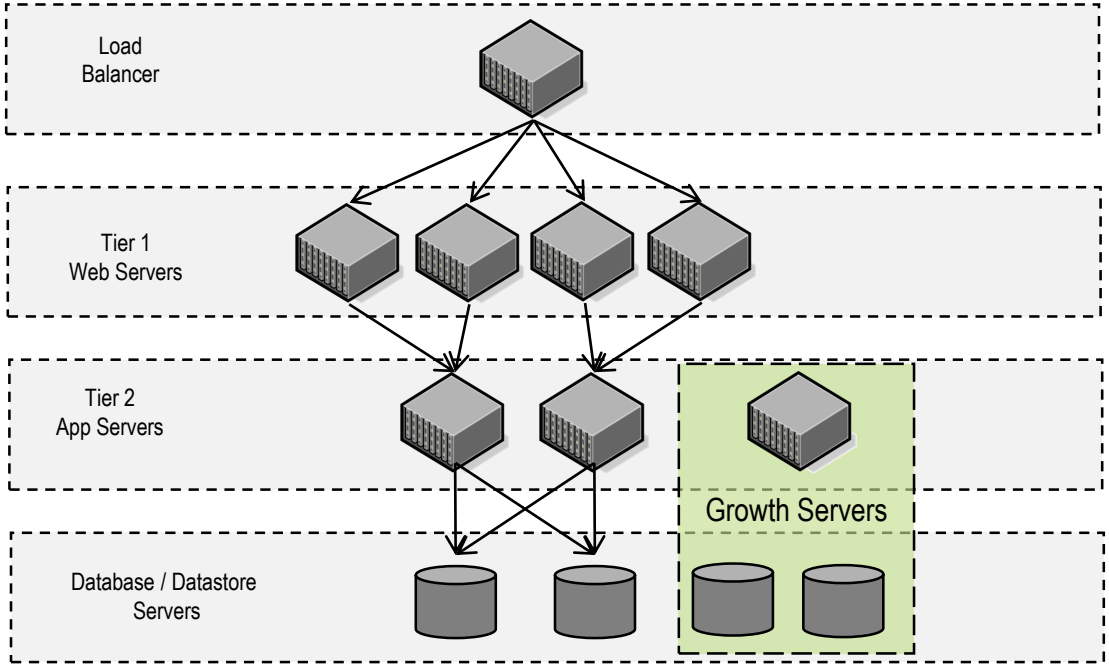
As usage would grow, previous hardware was swapped out whole for new larger hardware with "anticipated" new capacity.



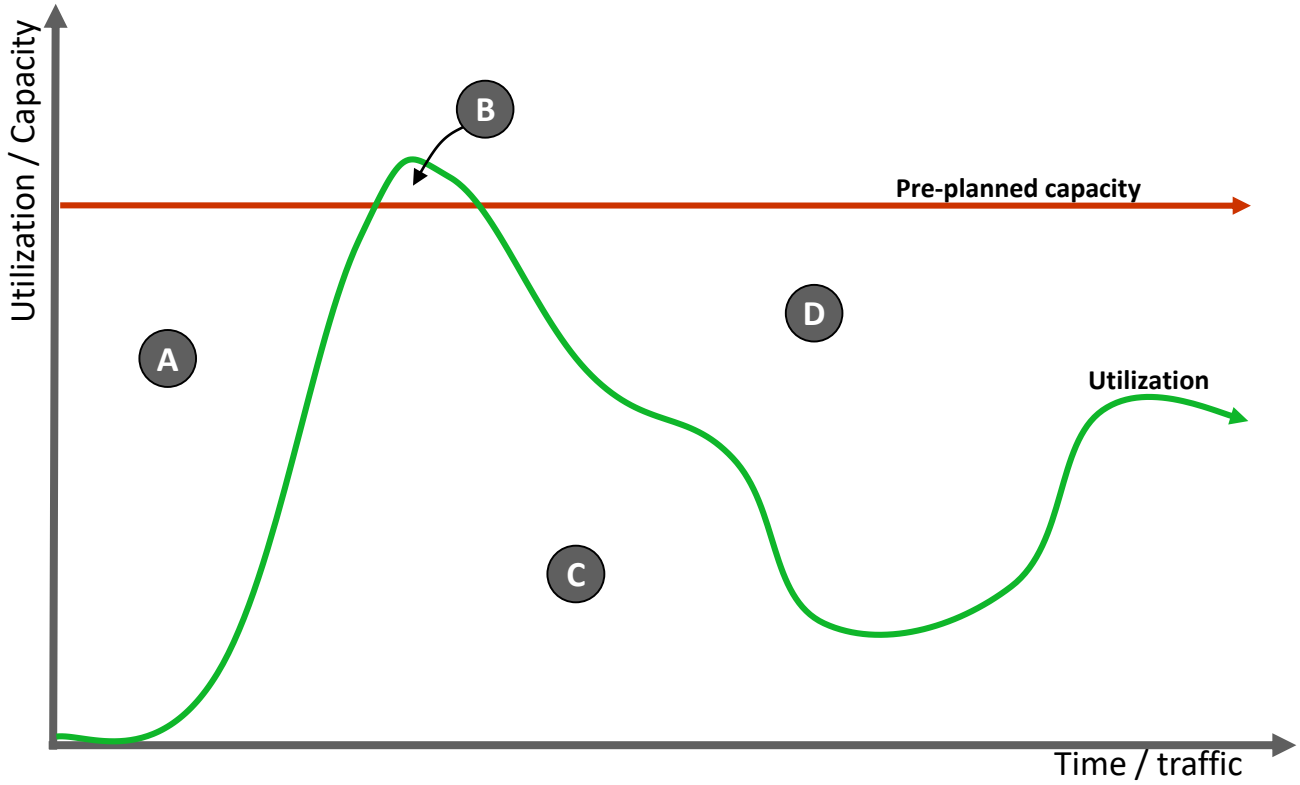
Cloud Scaling model = Elasticity

greatlearning

Modern applications can leverage IaaS or PaaS for scaling only the layers that demand it!

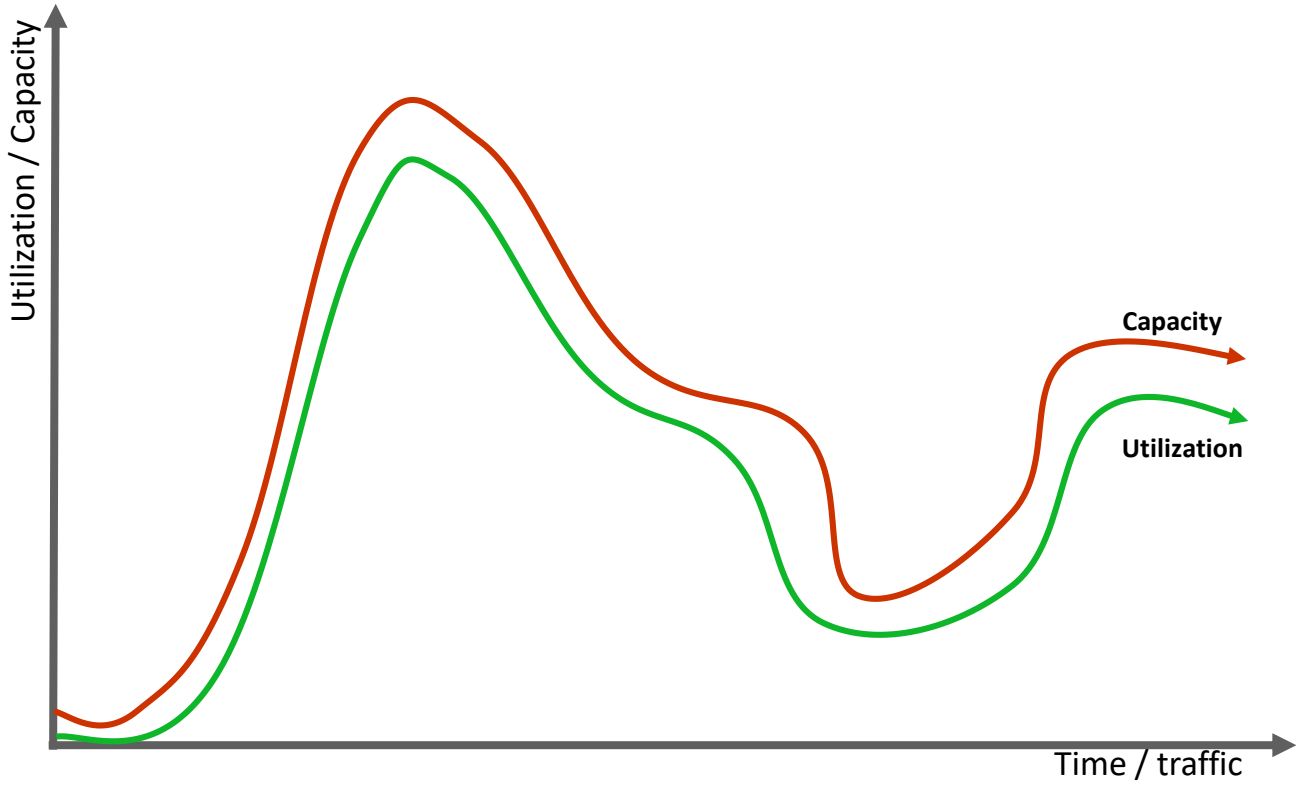


Cost economics - Classical model



What's happening in areas A, B, C & D? What about OPEX?

Cost economics - Cloud model



What happened now? Can you articulate the changes?

Scaling in Google & AWS

greatlearning

Total number of instances	Average QPS*	Average Latency*	Average Memory
37 total	0.177	339.5 ms	72.0 MBytes

Instances

QPS*	Latency*	Requests	Errors	Age	Memory	Availability
0.300	347.7 ms	227	0	0:08:29	74.6 MBytes	Dynamic
0.300	329.5 ms	247	0	0:08:24	69.8 MBytes	Dynamic
0.217	343.4 ms	212	0	0:03:27	70.1 MBytes	Dynamic
0.317	304.1 ms	234	0	0:03:25	75.0 MBytes	Dynamic
0.100	370.0 ms	227	0	0:03:24	71.2 MBytes	Dynamic
0.383	318.3 ms	218	0	0:03:22	70.7 MBytes	Dynamic
0.400	341.3 ms	230	0	0:03:13	71.0 MBytes	Dynamic
0.167	346.6 ms	100	0	0:02:26	67.2 MBytes	Dynamic
0.350	343.9 ms	67	0	0:01:35	66.6 MBytes	Dynamic
0.250	366.2 ms	38	0	0:01:15	65.6 MBytes	Dynamic
0.300	388.9 ms	57	0	0:01:33	66.0 MBytes	Dynamic
0.100	376.9 ms	37	0	0:01:18	66.1 MBytes	Dynamic
0.017	269.8 ms	78	0	0:03:20	65.5 MBytes	Dynamic

A script simulated load and Google spun up instances to handle it automatically.

AWS configuration allowing 1 min 4 max instances of certain type with specific rules of scalability

Scaling

Environment type: Load balanced, auto scaling

Number instances: 1 - 4

Scale based on Average network out

Add instance when > 6000000

Remove instance when < 2000000

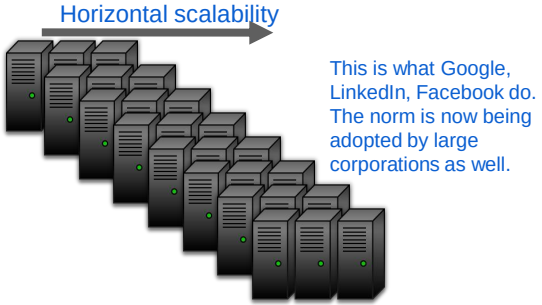
Instances

Instance type: t1.micro

Availability Zones: Any

Vertical/Specialized vs Horizontal/Commodity

VS



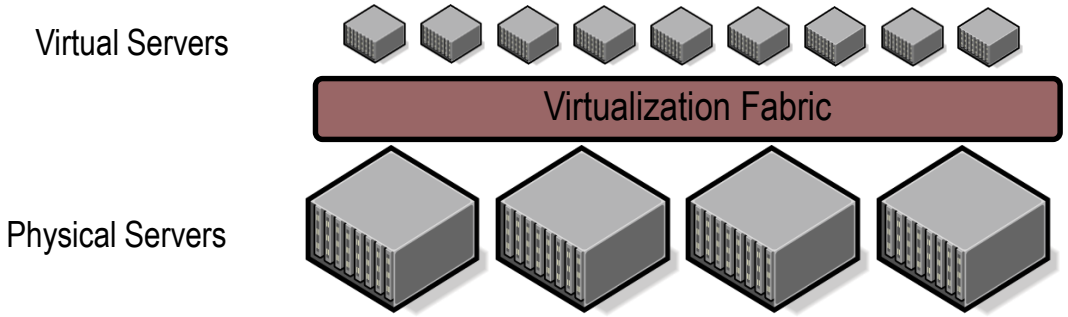
1. Large OPEX
2. Wasted/Idle resource
3. Failure takes out a large chunk
4. Expensive redundancy model
5. One shoe fitting all model
6. Too much co-existence

1. Low OPEX (rent on IaaS)
2. Maximum resource utilization
3. Failure takes out a small chunk
4. Inexpensive redundancy
5. Specific h/w for specific tasks
6. Very less to no co-existence

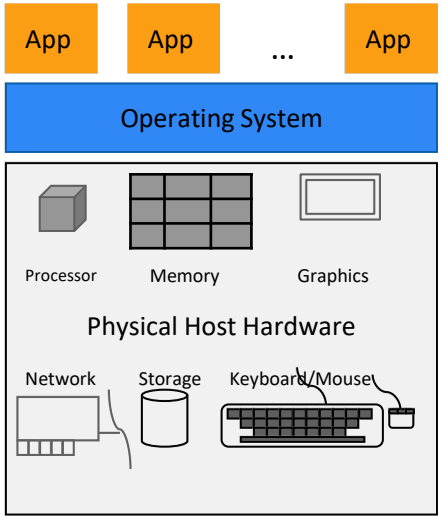
"Just in time" expansion; stay in tune with the load. No need to build ahead in time in anticipation and finally Horizontal scalability gives better "resilience"

Virtualization

- Virtualization of the computing resources, including servers, network, and storage, allows dynamic flexibility.
- Capacity can be more efficiently utilized.
- Quickly add new servers without delay due to procurement or installation.
- Easy to turn on or off virtual servers to handle scalability.
- Physical connectivity is done up front and configuration is done in software at provisioning time.
- Networking equipment and storage is virtualized as well.

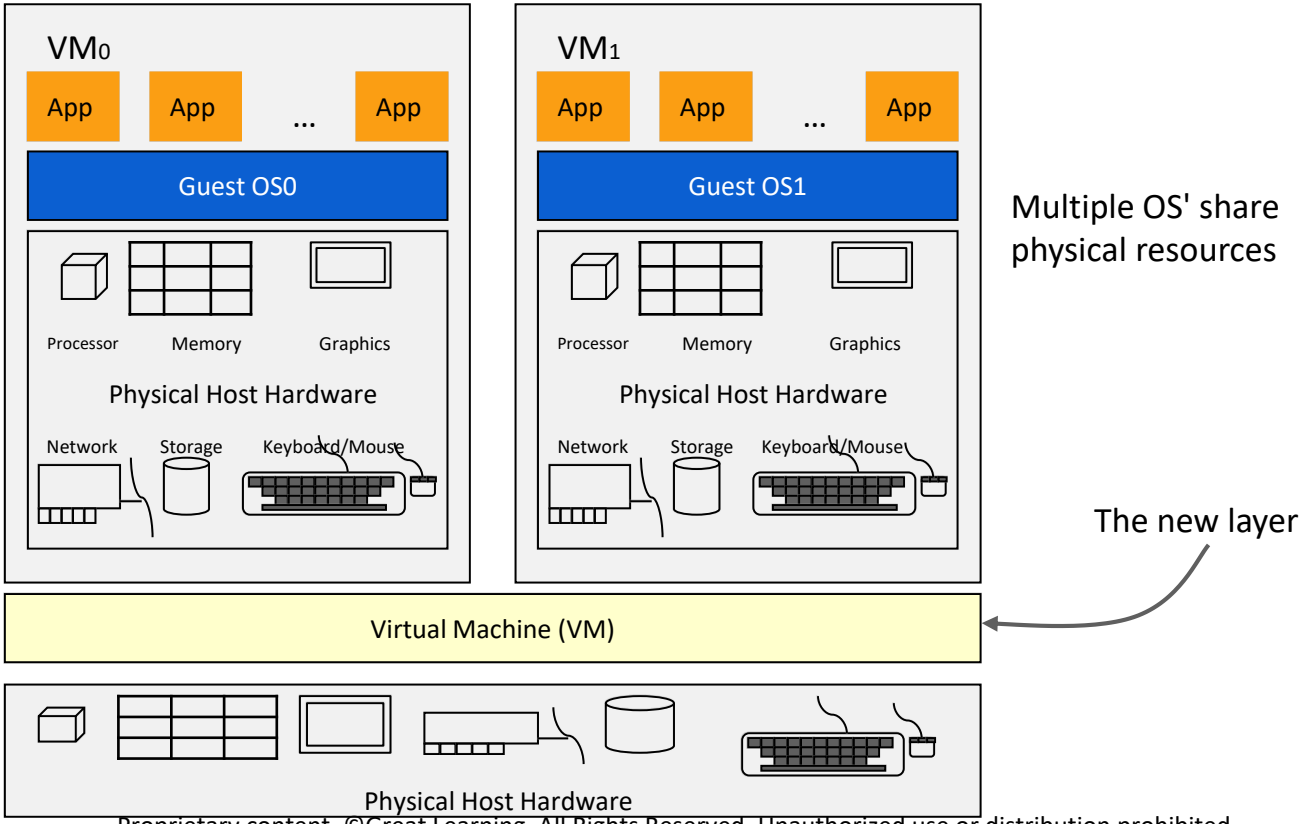


A typical application stack



A single OS owns all physical resources

Virtualized stack



Cloud VM characteristics

greatlearning

- Business Perspective (You get all of these ...)
  - Option of more computing resources available on demand
  - Elimination of upfront monitary commitment
  - Ability to pay for computing resources on a short term basis as need
  - There are options to acquire compute resources by "auction"
  - A service centric approach with self service & self managed

- Technology Perspective (... without doing any of these yourself!)
  - Partitioning: In virtualization, many applications and operating systems (OSes) are supported in a single physical system by partitioning (separating) the available resources.
  - Isolation: Each virtual machine is isolated from its host physical system and other virtualized machines. Because of this isolation, if one virtual-instance crashes, it doesn't affect the other virtual machines. In addition, data isn't shared between one virtual container and another.
  - Encapsulation: A virtual machine can be represented (and even stored) as a single file, so you can identify it easily based on the service it provides. In essence, the encapsulated process could be a business service.
  - Flexible: Should be able to configure and reconfigure to meet the growing and changing needs.

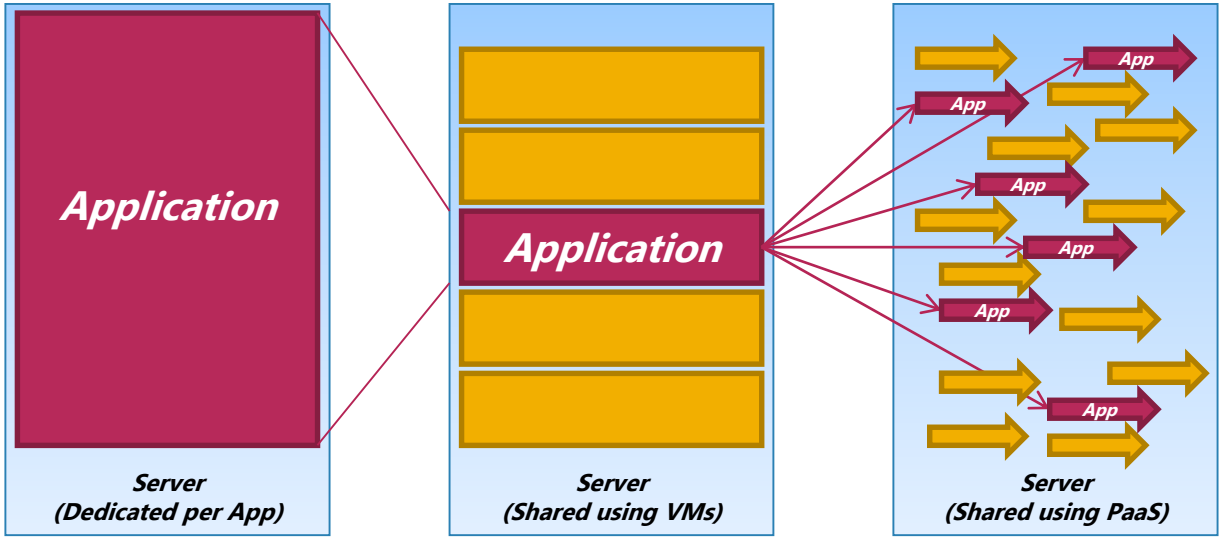
Virtualization drawbacks

- Amplified physical failures
- Skills required to setup perfectly
- Complex root cause analysis
- Estimating the number of VMs per physical hardware
- Performance reduces (extra layer)
- Some applications do not play well in a virtualized environment
- Licensing costs
- Hold on to these points

Evolving from IaaS to PaaS

VM allows hardware consolidation ...

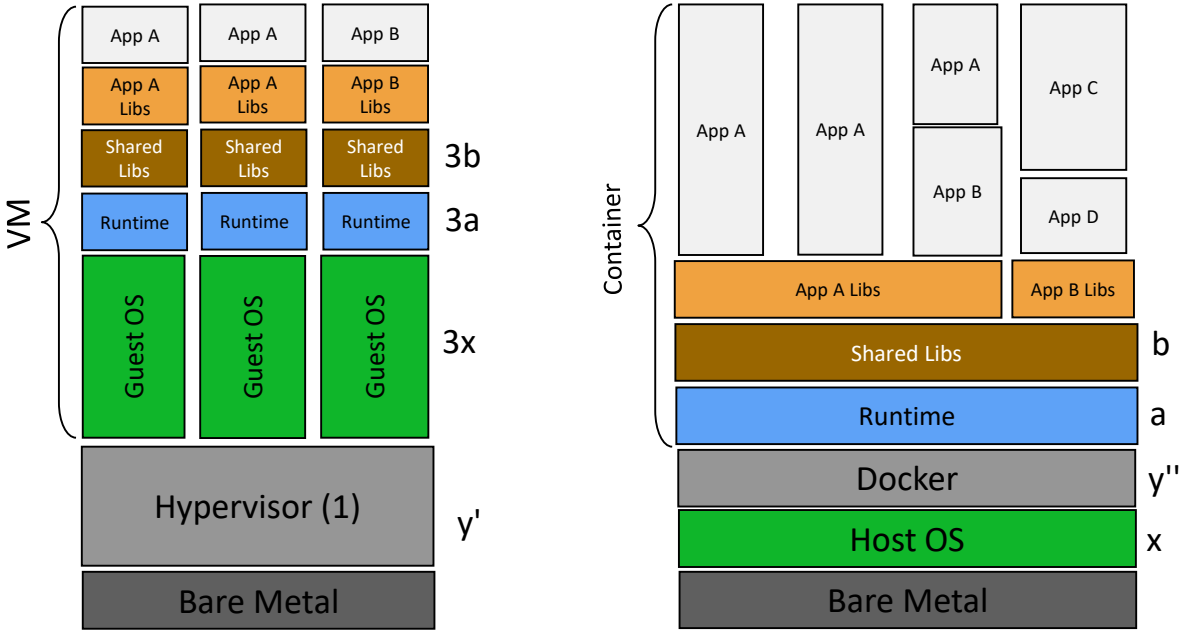
but what about the applications?



Next gen virtualization

- Given the virtualization drawbacks & the need for the dynamic swapping is virtualization the most appropriate strategy?
- Virtualization gives "pre-defined" buckets, is there something that can provide "on-demand" buckets?
- In the previous example we can deploy at max 5 applications and there is no guarantee that all resources will be fully utilized
- At the end of the day what's the real purpose of virtualization?
  - Run more applications
  - Give applications the environment it deserves
  - Create a sandbox for each so that no one steps on each other
- Enter "Container"
  - Application focused bundling
  - Common aspects (os, lib et al) are shared
  - Smaller footprints
  - Quick load time
  - Also known as LXC (LinuX Containers)
- Container based virtualization is the contemporary option being adopted by most PaaS cloud players

Container vis-a-vis virtualization



y' >> y''

Faceoff - Container VM vs Metal VM

greatlearning

- All apps need to be deployed on the same OS image (good choice for PaaS)
- VM can have a choice of OS (good option for IaaS)
- Containers have less isolation among them
- VM allows for higher isolation
- Containers will usually take milli-seconds to start
- VM may take minutes

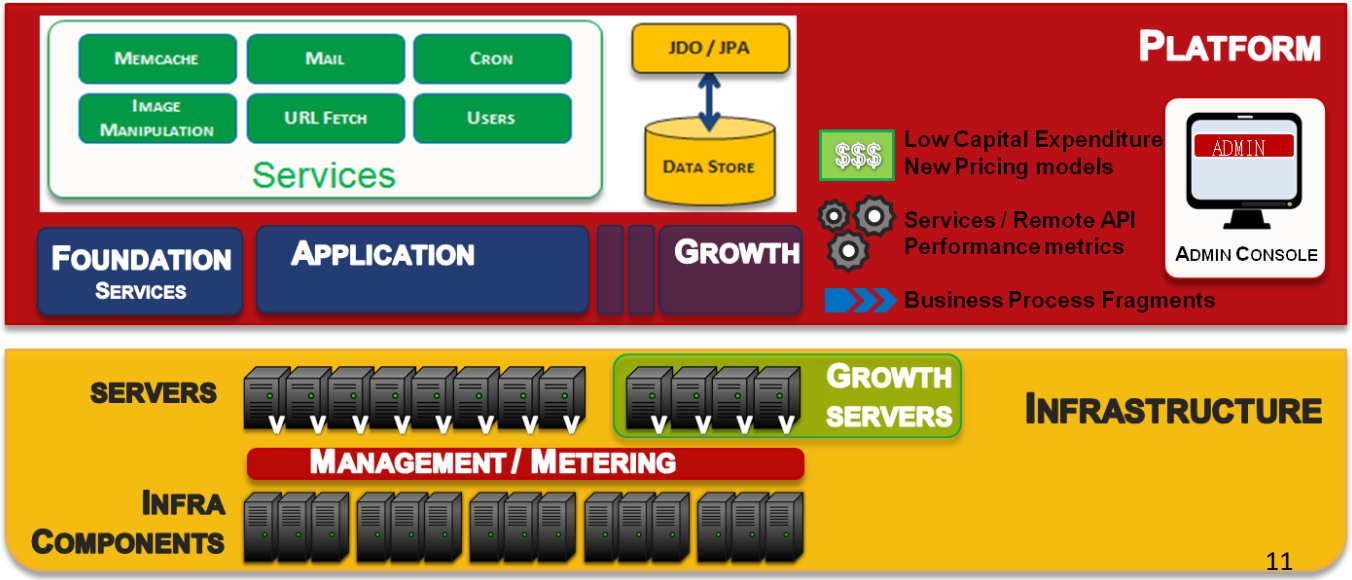
Container VM OR Metal VM?

greatlearning

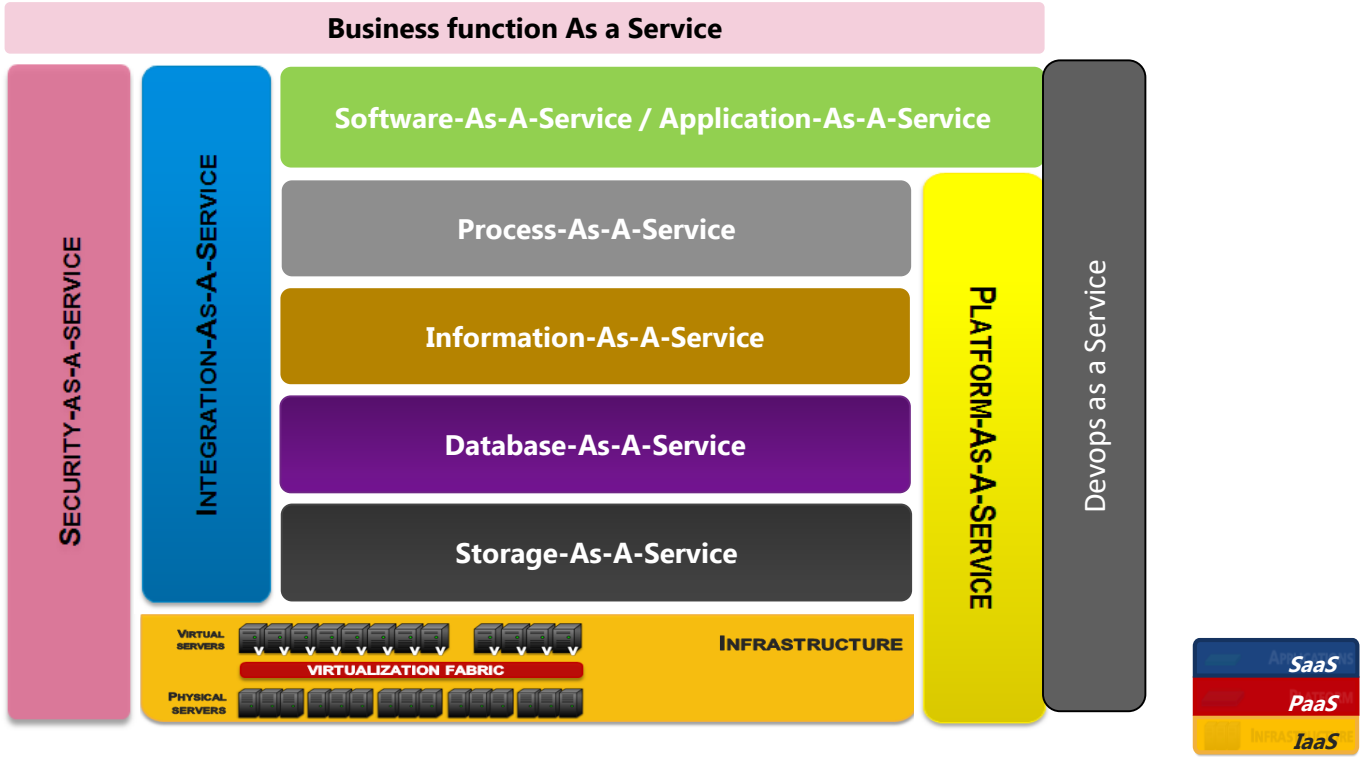
- Is the question correct? Should it be AND/OR?
- Chances are you need both
- IaaS will require VMs on bare metal
- Specific solutions requiring only runtimes may opt for containers
- It all depends on the architecture and the nature of the problem you are looking to solve
- Enough of VMs, let's get a move on ...

PaaS overview

- Needs an operational IaaS to be "effective"
- Abstracts the infrastructure layer completely
- Provides monitoring at two levels
  - Infrastructure
  - Application



Cloud services taxonomy



Price economics & decision making

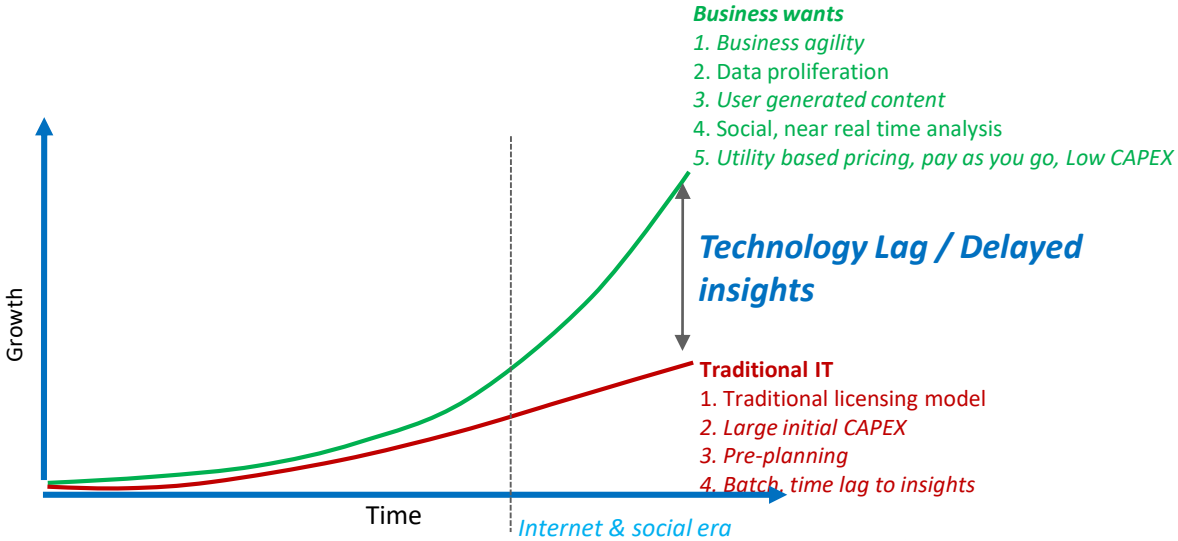
- Accounting says: Profit = Revenue - Costs
- Cloud allows you to control and monitor the ops costs
- What about pricing model? How can cloud influence it?
- Let's see how price is arrived at
  - Cost based pricing
  - Value based pricing
- Cost based
  - Cost of building product
  - Cost of maintaining the product
  - Providing support for the product
  - Other factors that may be specific to the product as well as the organization
  - Charging for services offered and/or enhanced
  - Pass through costs can find a way in the product price
- Value based
  - Customer perception about the product
  - What values customer derives from the product
  - Longevity of the product
  - Recognizable name associated with luxury

Price economics & decision making

- Can the price of the product be constant/static?
- Does information velocity have a bearing on the rate of price change?
- How can we maximize margins?
  - Alteration to price, popularity based etc
  - Cost of dependent services/api do not remain constant
  - Get analytics quickly
  - Reliability of data and its availability
  - Accessibility of information anytime/anywhere for agile business decisions
  - Accessing business applications to take quick actions
- To get to all of the above we need "Information velocity" ... next

Data to Information xform velocity

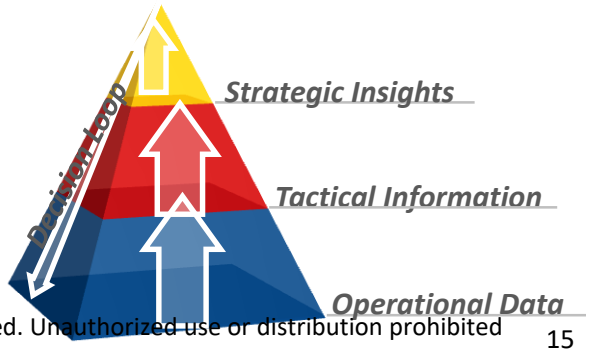
greatlearning



To stay several steps ahead of the competition organizations all over the world are asking a simple question

"Can we get some insights out of our data? Quickly please!"

We need "Big Data & Analytics"



Challenges with Distributed Computing

- Heterogeneity
- Fault handling
- Consistency
- Global concurrency
- Upgrades and maintenance
- Local resources - file system
- Application sessions & transient data
- Clock synchronization

Designing applications for cloud

- Evaluate enterprise architecture impact
- Choice of the right service provider
- Defining the business process and integration touch points
- Security fabric cutting across all providers
- Think stateless, adapters, foundation
- Micro service approach

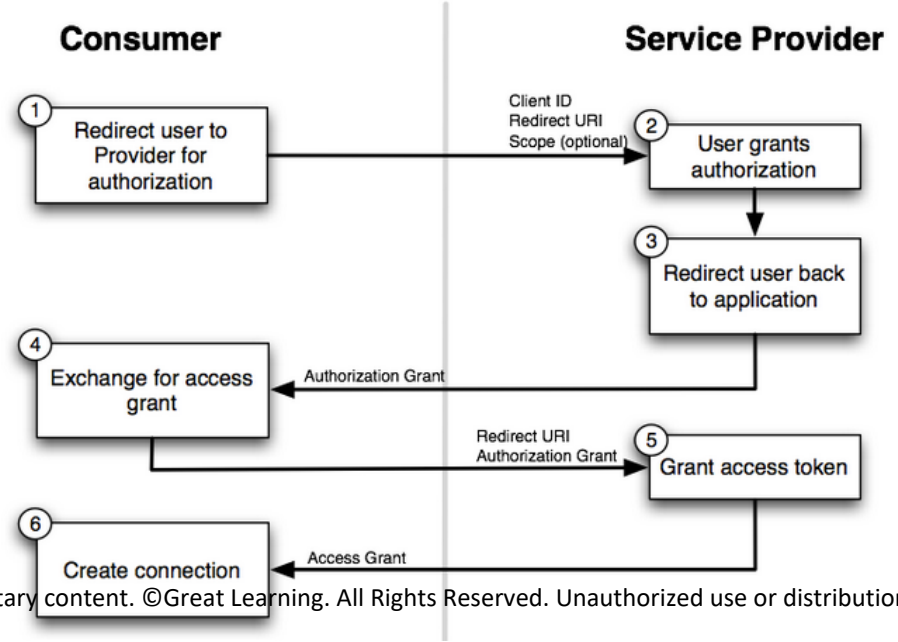
Distribution of Control between Service Models

- Decentralized Administration
  - principle of local autonomy, which implies that each service model retains administrative control over its resources
- Secure Distributed Collaboration
  - Due to the heterogeneous nature of the cloud, resource and service policies might use different models requiring seamless interoperation among policies (SLA)
- Credential Federation
  - decentralized single-sign-on mechanism
- Placement of functionality
  - Right provider for the functionality needed in the business process
- Federated Data Collaboration
  - In an interleaved business process it is imminent that data payload is managed
- Loose coupling
  - Services are owned by different providers with their own evolution lifecycle and versioning

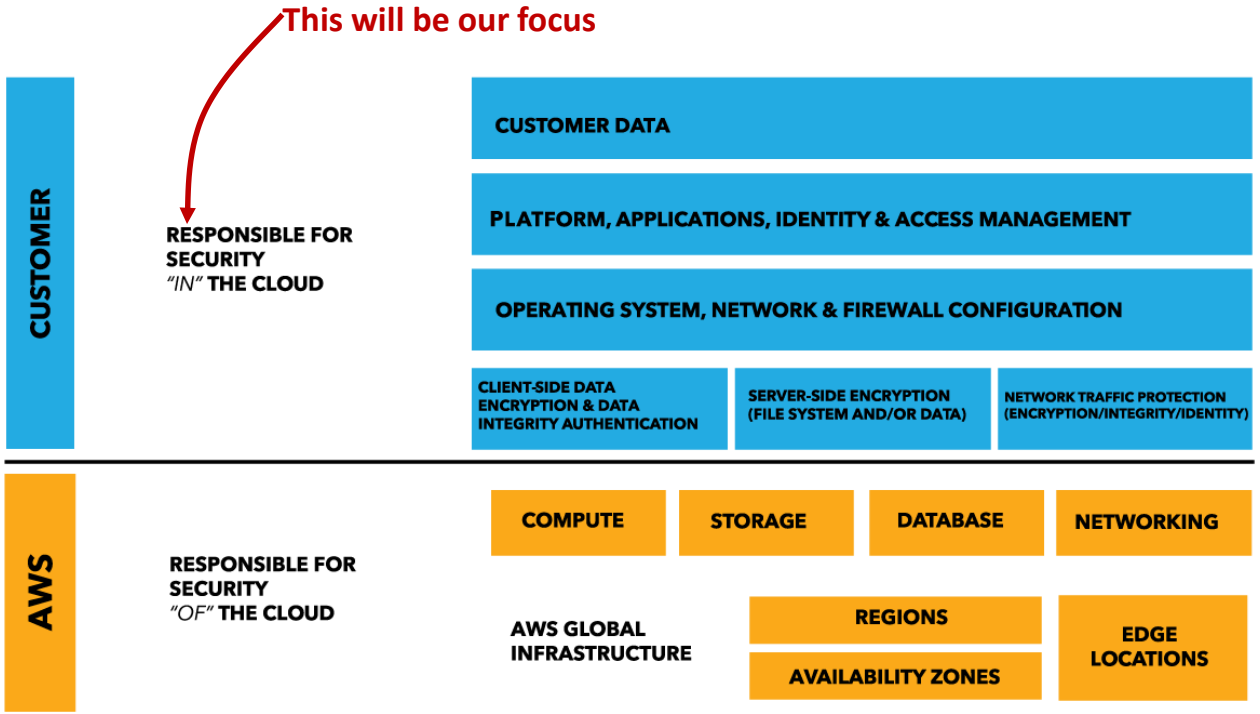
Security fabric for the architecture with multiple providers

greatlearning

- The usual model in the cloud is oAuth
- The current version is 2.0
- Here is a typical flow to integrate with Facebook authentication



"Shared responsibility" model



https://aws.amazon.com/compliance/shared-responsibility-model/

Diversity of programming languages

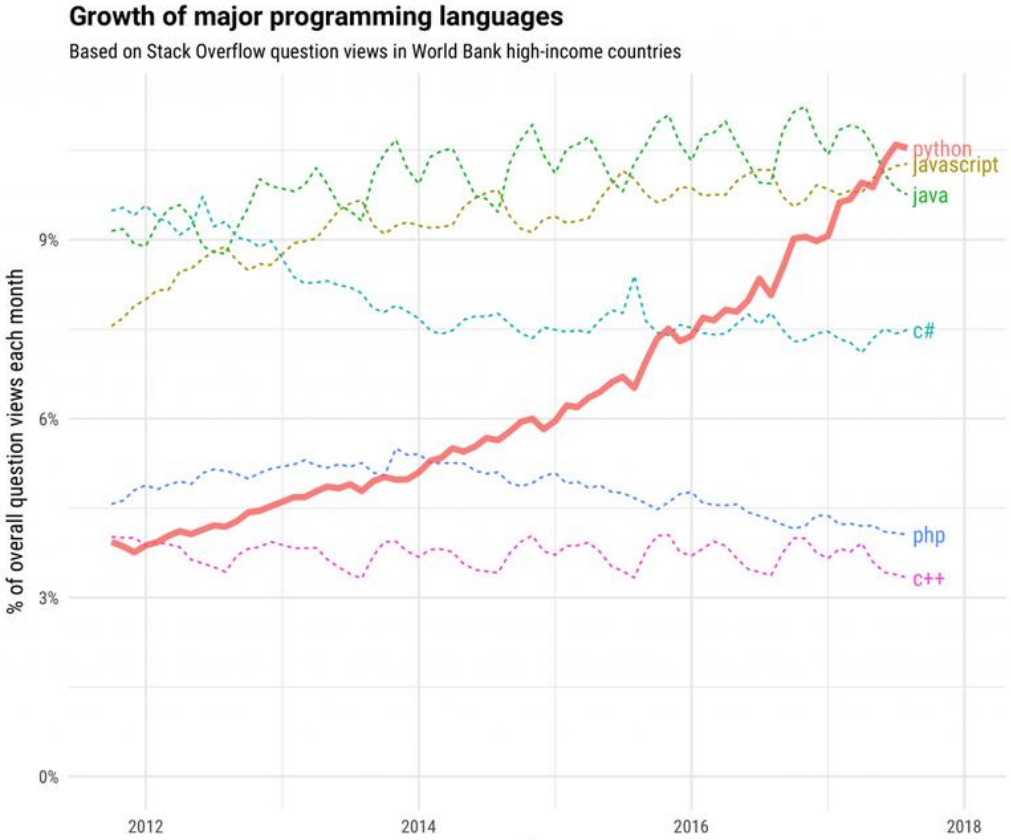
greatlearning

- Machine language
- Procedural
- Data oriented
- Object oriented
- Functional

- In addition there are programming models that are gaining traction
  - Non blocking
  - Actor oriented programming
  - Concurrent environments
  - Reactive programming

- Examples
  - Boss thread and worker thread model
  - Event oriented programming in the server!
  - Leveraging more than 1 CPU/cores by the VM
  - In MS Excel if a1=b1+c1 (if b1 changes a1 auto changes)

Language popularity index



http://www.techrepublic.com/article/which-is-the-fastest-growing-programming-language-hint-its-not-javascript/

Infrastructure automation

- We are looking at a very complex infrastructure with dozens or 100' of multi-tiered applications
- Cloud provider operations as well as inhouse IT management is not getting easier
- Solution is
  - Chef.io
  - Puppet.com
  - Ansible.com
  - CI using Jenkins

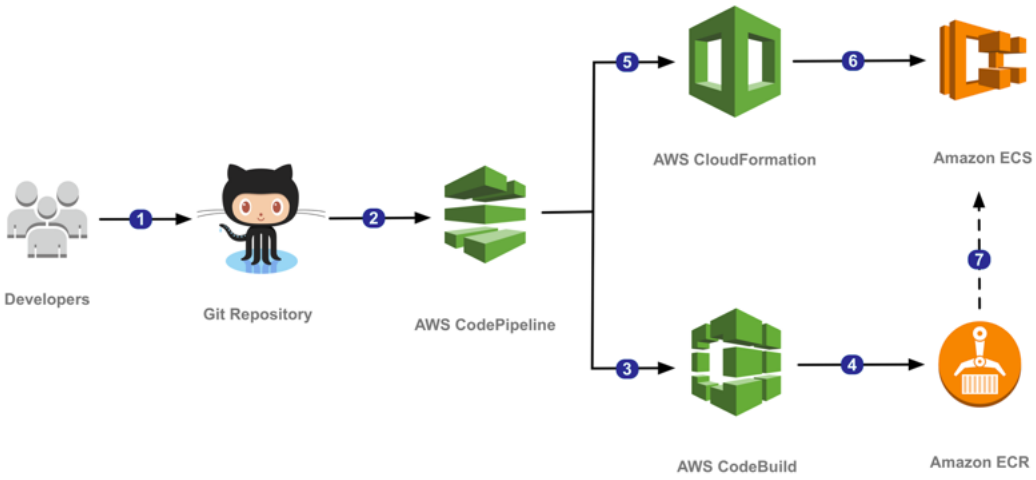
- http://www.infoworld.com/article/2609482/data-center/data-center-review-puppet-vs-chef-vs-ansible-vs-salt.html

Continuous Integration and code deployment models

greatlearning

- Central code repositories
  - SVN - classical model
  - GIT - more contemporary and allows for more parallel development
- Auto build process - create "Machine images" or "Container images"
  - Trigerred at the time of checkin
  - Scheduled build
- Deployment from repositories to
  - Dev, Stage, Prod
  - Auto update app parameters that may be different in different environments
  - Labeling
- Dispatching emails upon success / failure
- Many products
  - Jenkins (eg Cloudbees)
  - Chef, Puppet, Ansible, Salt

Cloud CI/CD



1. Developers continually integrate their changes together into a main branch hosted within a source code repository system such as GitHub.
2. AWS CodePipeline polls the source code repository and triggers an execution of the continuously delivery pipeline when a new revision is found.
3. AWS CodePIpeline sends the new revision to AWS CodeBuild which builds a Docker container image from the source code.
4. AWS CodeBuild pushes the newly built Docker container image tagged with the build ID to an Amazon ECR repository.
5. AWS CodePipeline initiates an update of the AWS CloudFormation stack which defines the Amazon ECS task definition and service.
6. AWS CloudFormation creates a new task definition revision referencing the newly built image and updates the Amazon ECS service.
7. Amazon ECS fetches the new container from Amazon ECR and replaces the old task with the new one which completes the deployment.

amazon web services

Internet of Things (IoT)

The Internet of Things (IoT) is the network of physical objects or "things" embedded with electronics, software, sensors and connectivity to enable it to achieve greater value and service by exchanging data with the manufacturer, operator and/or other connected devices.

-Wikipedia

How do these work?

- Interactive smart tables
- Smart refrigerators
- Smart air conditioners
- Mood based Ambiance
- Smart water heaters
- Control window shades from IP phones
- Mars robots!

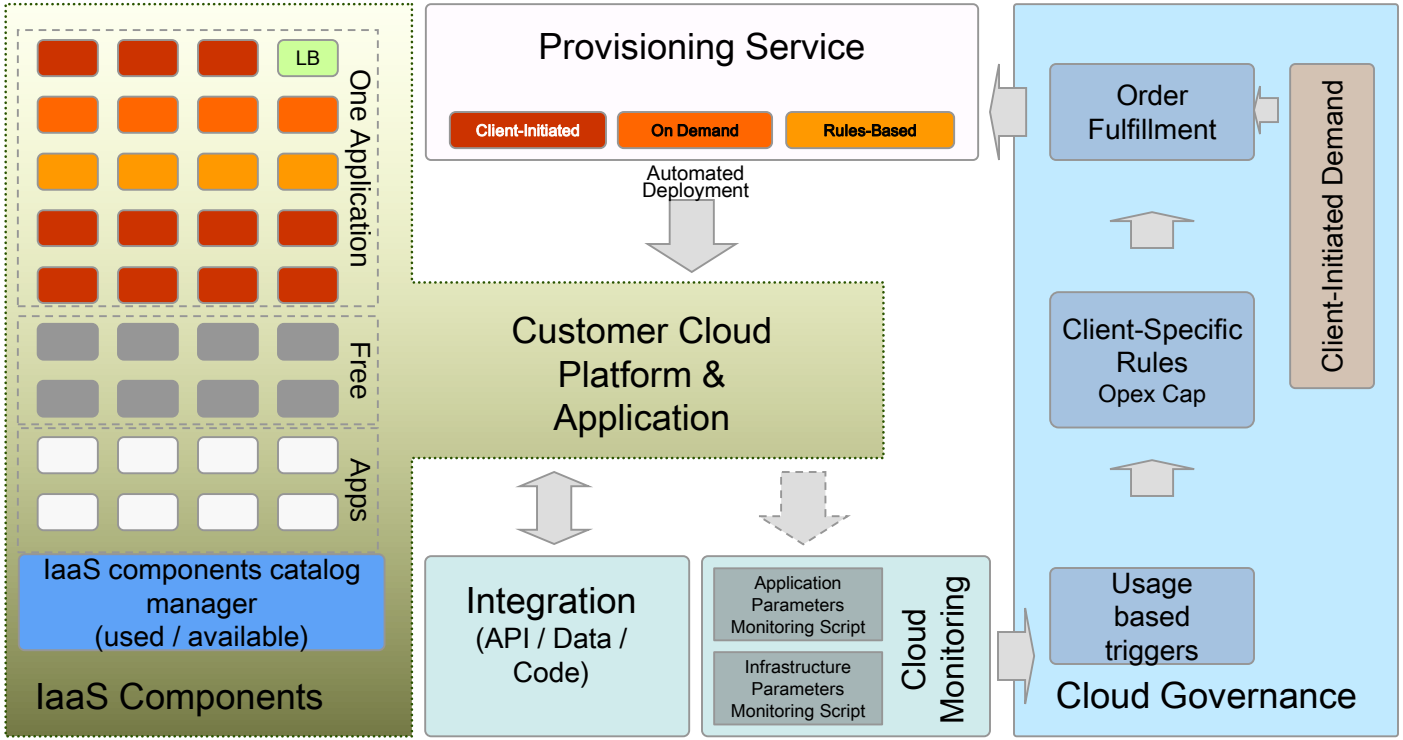
I Robot

Frameworks for abstraction

Google Brillo Website https://developers.google.com/brillo

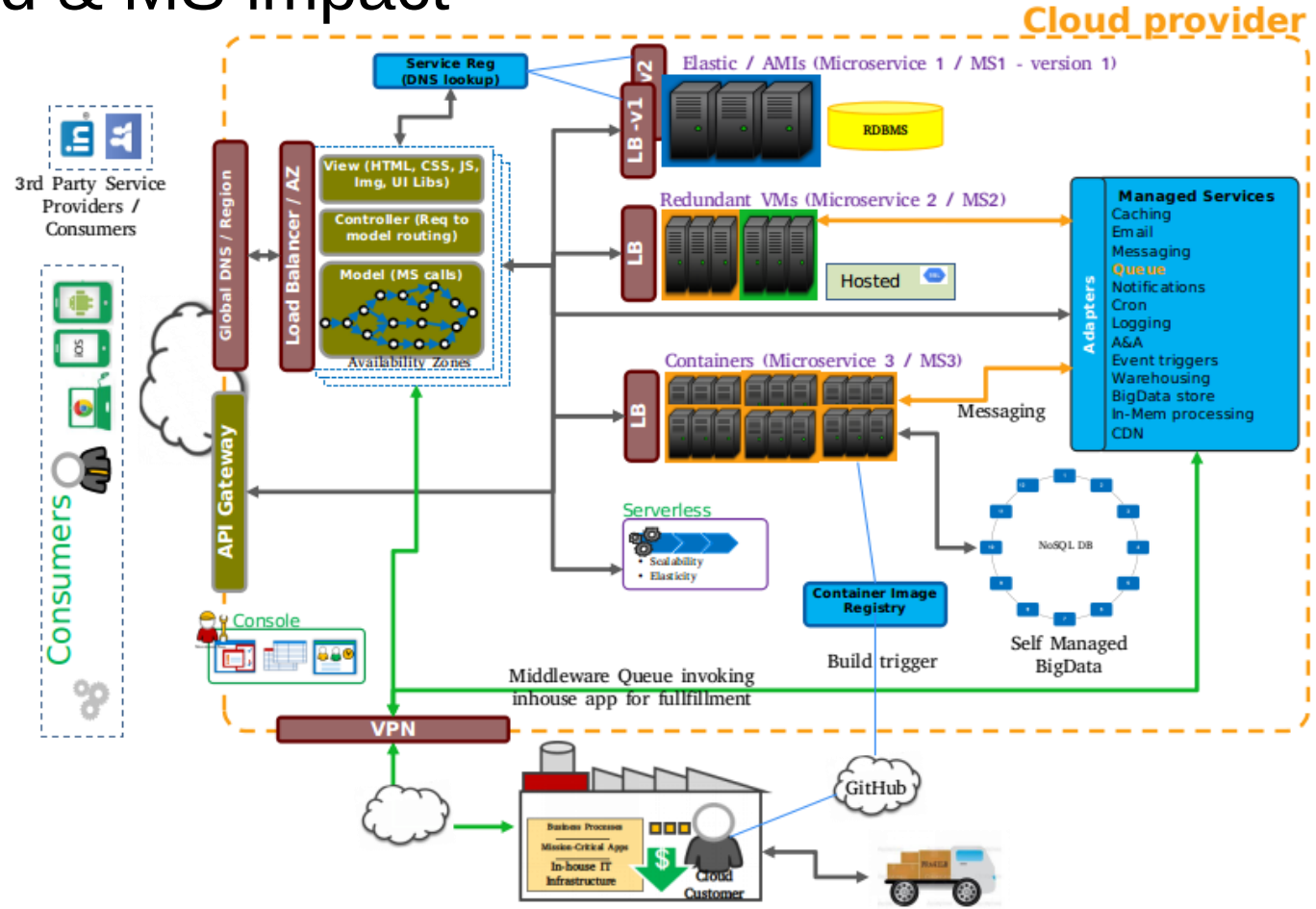
Overview https://www.youtube.com/watch?v=2rPkbyyviGI
Weave https://www.youtube.com/watch?v=uIIZD4KuIJM

https://aws.amazon.com/greengrass/



The processes that monitor the health of the components are not represented. They are internal to the IaaS provider but can trigger provisioning and/or maintenance activity.

Cloud & MS impact



Cloud computing concerns

- Progressive Architecture Development is where the overall architecture vision is broken into piece meal and developed over a period of time. The cloud approach should follow a similar approach.
- This is particularly important because the cloud applications space is still maturing barring a few select players. The following are the typical impediments perceived by organizations when it comes to cloud application implementations.

- *Current enterprise apps can't be migrated conveniently*
- *Lock in with proprietary architecture*
- *Risks – Legal, regulatory, and business*
- *Difficulty of managing cloud applications*
- *Lack of clarity of SLA ownership*
- *Unclear ROI*

What next?

Closing thoughts

greatlearning

"Cloud computing as a concept glues together several other, often independent, concepts in a complimentary way to open up new operating models and opportunities for various industries.

Cloud Computing has evolved from, and extends, several concepts that have been around for some time, such as SaaS, utility Computing, Grid Computing, Virtualization, Real-Time Infrastructure, Web Platforms, and SOA."