Desmistificando Microsserviços e DevOps: Projetando Arquiteturas Efetivamente Escaláveis

Prof. Vinicius Cardoso Garcia
vcg@cin.ufpe.br :: @vinicius3w :: assertlab.com

[IF1004] - Seminários em SI 3

Licença do material

Este Trabalho foi licenciado com uma Licença

Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 3.0 Não Adaptada

Mais informações visite

http://creativecommons.org/licenses/by-nc-sa/3.0/deed.pt

AssertLab
Advanced Software and Systems Engineering Research Technologies

Centro de Informática

40 anos

Background

The Cloud as Platform

"We've redefined cloud computing to include everything that we already do. ... The computer industry is the only industry that is more fashion-driven than women's fashion. ... We'll make cloud computing announcements because if orange is the new pink, we'll make orange blouses. I'm not going to fight this thing."

— Larry Allison (Excerpt From: Bass, Len. "DevOps: A Software Architect's Perspective")

Cloud Computing

- The National Institute of Standards and Technology (NIST) has provided a characterization of the cloud with the following elements
  - On-demand self-service
  - Broad network access
  - Resource pooling
  - Rapid elasticity
  - Measured service

– U.S. National Institute of Standards and Technology, Special Publication 800-145

Cloud Service Models

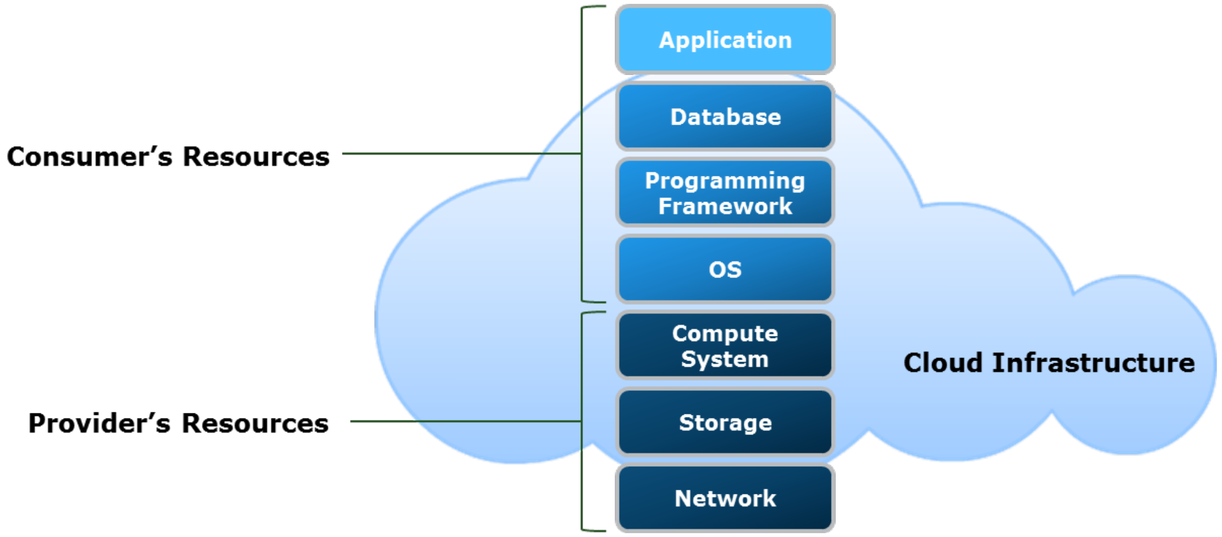
- A cloud service model specifies the services and the capabilities provided to consumers

- NIST specifies three primary cloud service models:
  - Infrastructure as a Service (IaaS)
  - Platform as a Service (PaaS)
  - Software as a Service (SaaS)

Infrastructure as a Service

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components , (e.g., host firewalls).

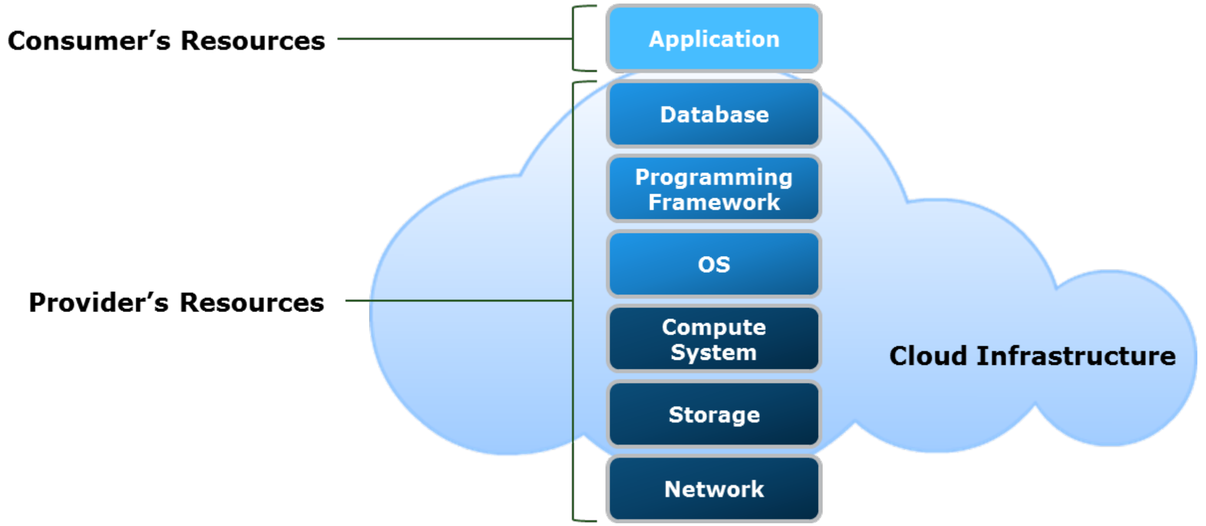
– U.S. National Institute of Standards and Technology, Special Publication 800-145



Platform as a Service

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

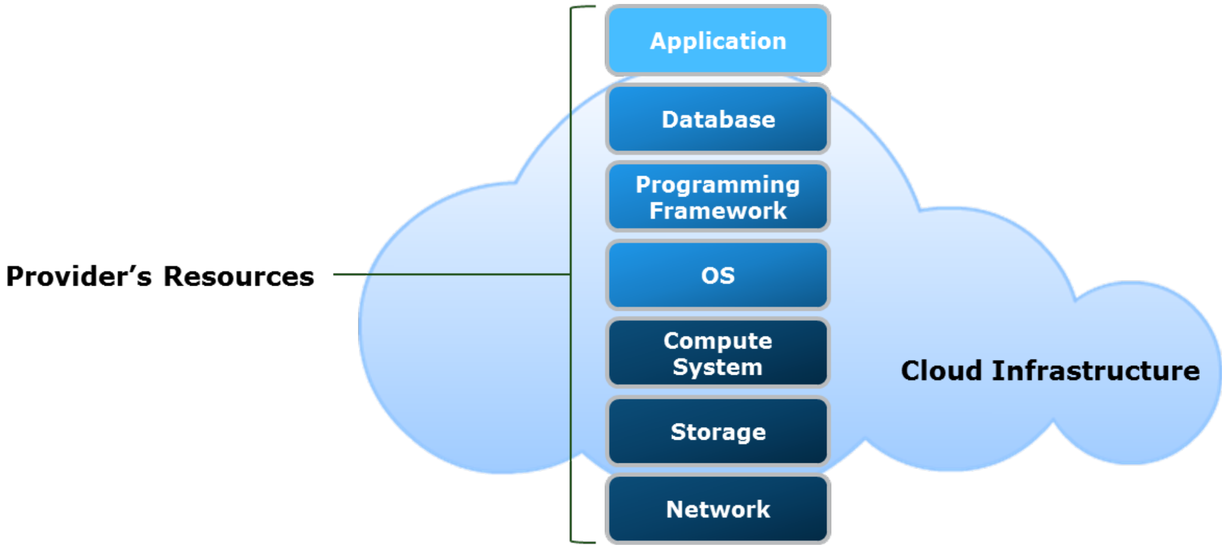
– U.S. National Institute of Standards and Technology, Special Publication 800-145



Software as a Service

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser, (e.g., web-based email, or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

– U.S. National Institute of Standards and Technology, Special Publication 800-145

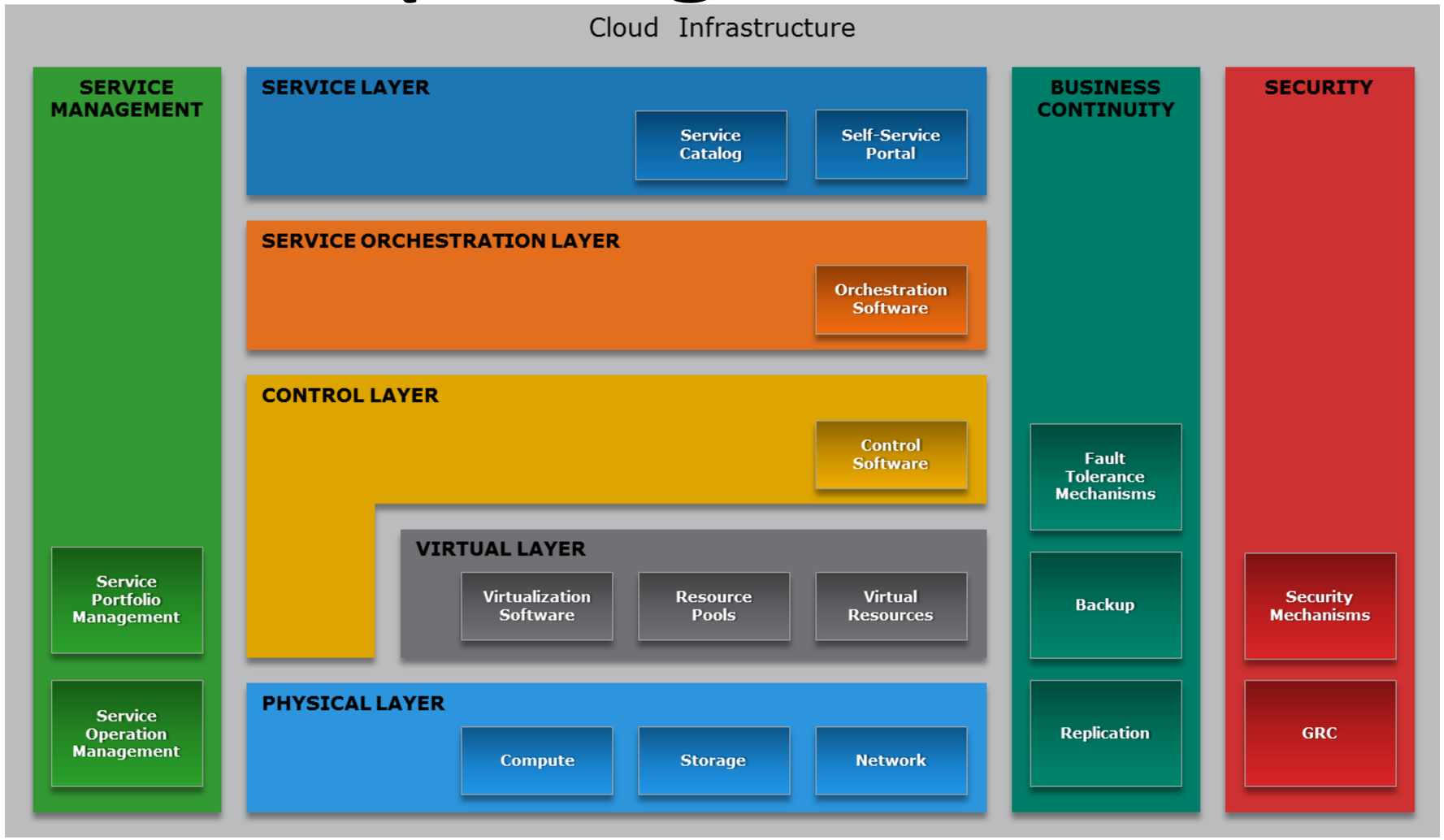


Cloud Deployment Models

- A cloud deployment model specifies how a cloud infrastructure is built, managed, and accessed

- NIST specifies four primary cloud deployment models:
  - Public
  - Private
  - Community
  - Hybrid

Cloud Computing Reference Model



– U.S. National Institute of Standards and Technology, Special Publication 500-292

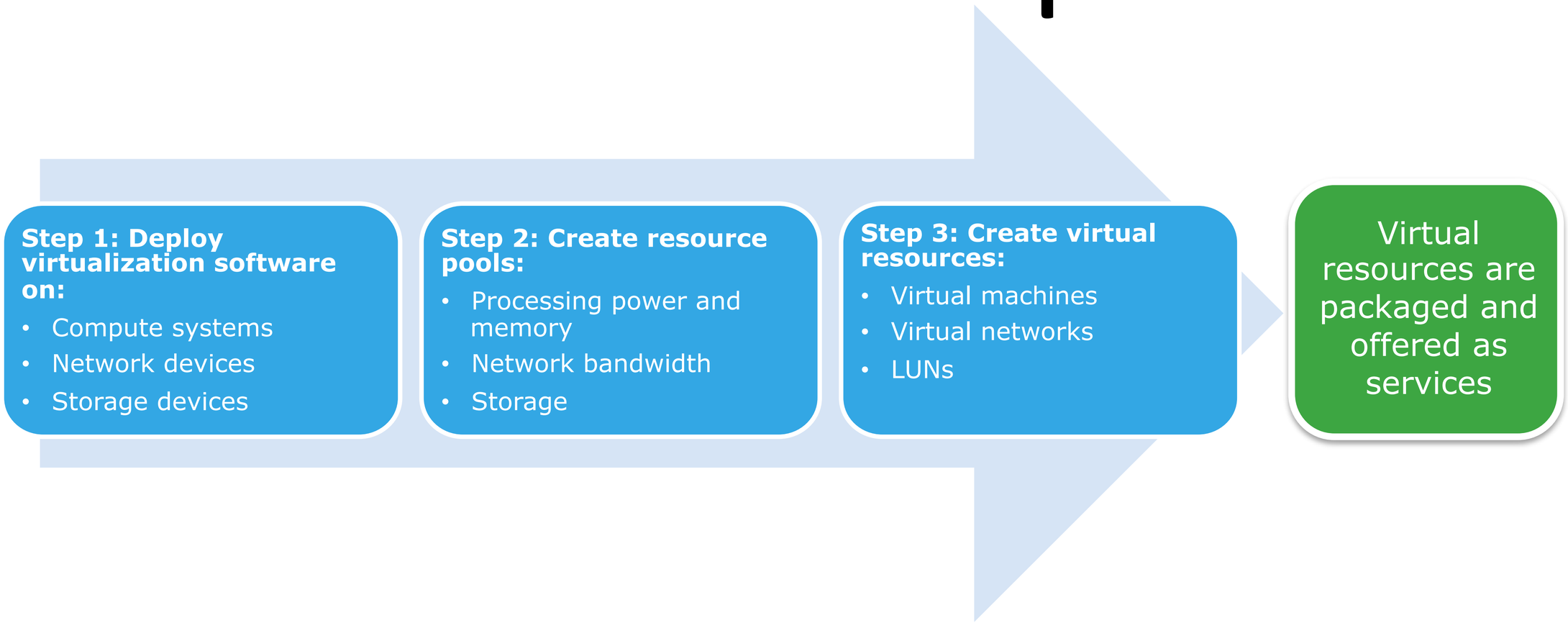
Features of the Cloud

Virtualization

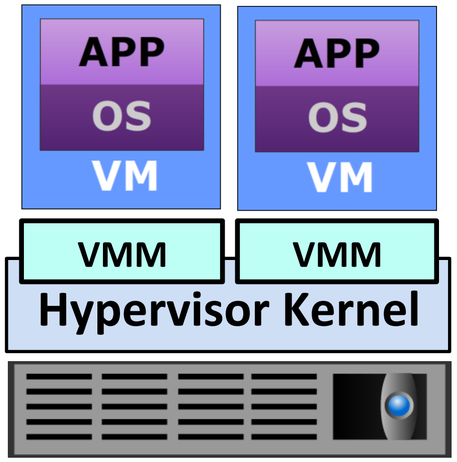
- Refers to the logical abstraction of physical resources, such as compute, network, and storage that enables a single hardware resource to support multiple concurrent instances of systems or multiple hardware resources to support single instance of system.

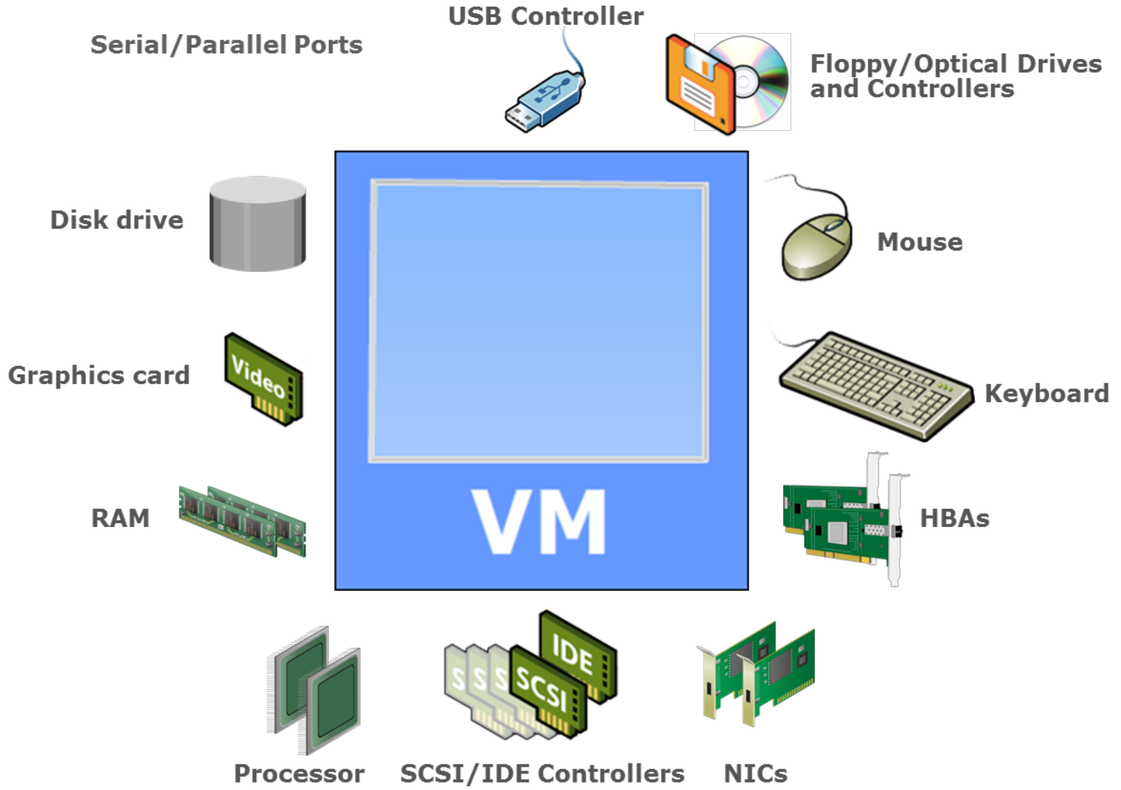
- Enables a resource to appear larger or smaller than it actually is
- Enables a multitenant environment improving utilization of physical resources

Virtual Process and Operacions

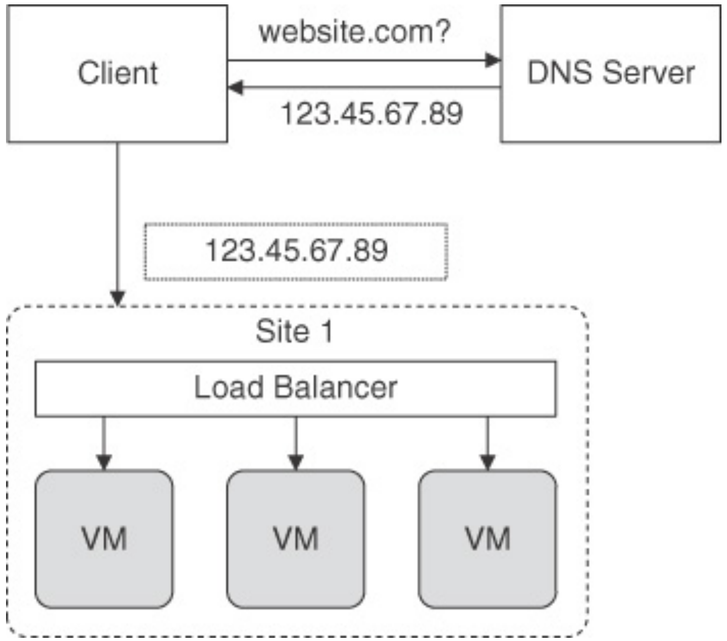


Virtualization Elements

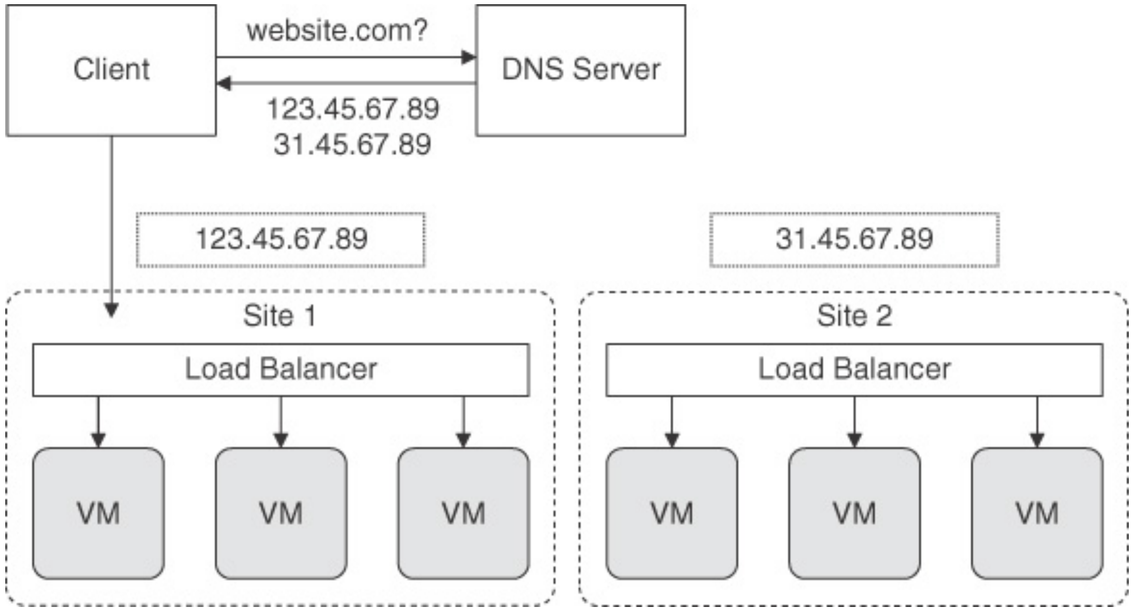




IP and Domain Name System Management



DNS returning an IP address



DNS returning two addresses for a single URL

Platform as a Service (PaaS)

- Many of the aspects we discussed so far are IaaS-specific.
  - PaaS services reside at a higher level of the stack and hide underlying details to a degree
- PaaS allow you to run applications in predefined environments
  - a set of core services (e.g., hosting of Java web apps, Ruby Gems, Scala apps, etc.)
  - catalogue of add-ons (e.g., specific monitoring solutions, autoscaling options, log streaming, and alerting, etc.)"

PaaS & Ops Environments

- Similar to some of the services offered by traditional Ops departments

- However, you have more add-ons and newer options more quickly than in traditional Ops departments.

Distributed Environment

- Time
  - what data to maintain in memory or on the disk is a critical performance decision.
  - where persistent data is physically located will also have a large impact on performance
- Failure
  - Amazon released some data stating that in a datacenter with ~64,000 servers with 2 disks each, on average more than 5 servers and 17 disks fail each day.”

Failure

- A list of problems arising in a datacenter in its first year of operation (from a presentation by Jeff Dean, Google)
  - ~0.5 overheating (power down most machines in <5 minutes, ~1–2 days to recover)
  - ~1 PDU failure (~500–1,000 machines suddenly disappear, ~6 hours to come back)
  - ~1 rack-move (plenty of warning, ~500–1,000 machines powered down, ~6 hours)
  - ~1 network rewiring (rolling ~5% of machines down over 2-day span)
  - ~20 rack failures (40–80 machines instantly disappear, 1–6 hours to get back)
  - ~5 racks go wonky (40–80 machines see 50% packet loss)
  - ~8 network maintenances (4 might cause ~30-minute random connectivity losses)
  - ~12 router reloads (takes out DNS for a couple minutes)
  - ~3 router failures (have to immediately pull traffic for an hour)
  - ~dozens of minor 30-second blips for DNS
  - ~1,000 individual machine failures
  - ~thousands of hard drive failures
  - slow disks, bad memory, misconfigured machines, flaky machines, etc.
  - long-distance links: wild dogs, sharks, dead horses, drunken hunters, etc.

any particular VM or portion of a network may fail

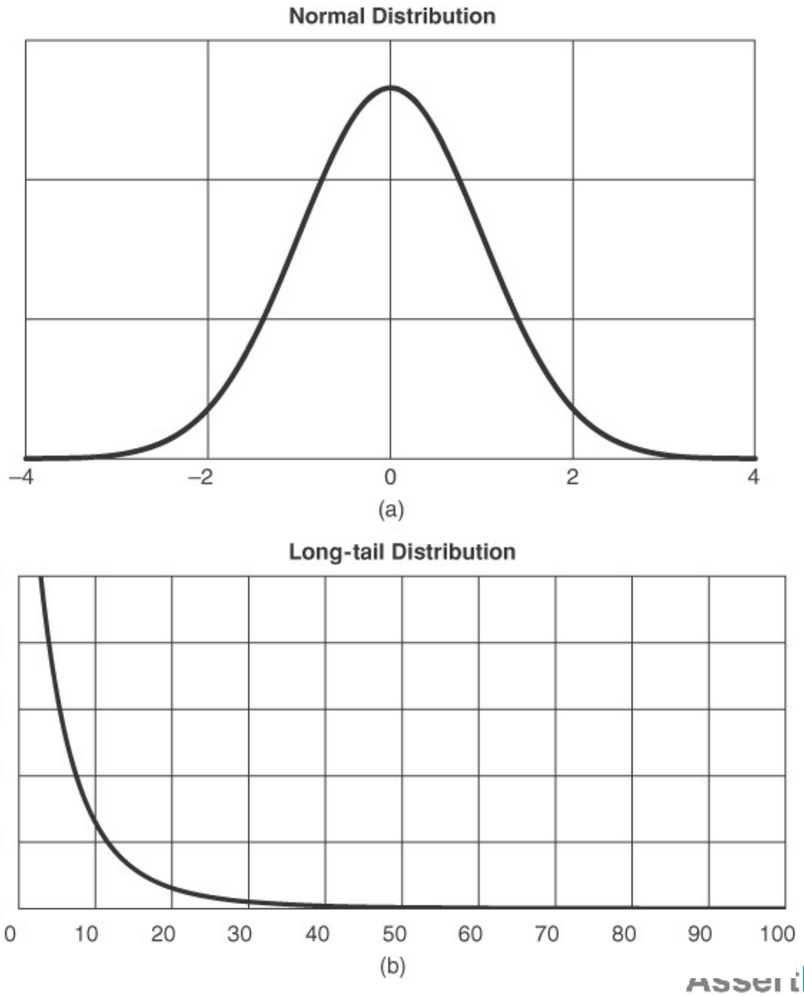
Failure of a VM

- If a stateless component fails, it can be replaced without concern for state. On the other hand, state must be maintained somewhere…
  - A stateless component: If a VM is stateless, then failure of a VM is recovered by creating another instance
  - Client state
  - Application state contains the information specific to an application or a particular user of an application
    - Small amounts of persistent state ~> ZooKeeper or Memcached
    - Moderate amounts of persistent or semi-persistent state ~> Memcached
    - Large amounts of persistent state ~> HDFS

The Long Tail

- In the cloud, many phenomena such as response time to requests show a long-tail distribution

- This result is often due to the increased probability of failure with more entities involved, and the failure of one component causes response time to be an order slower than usual

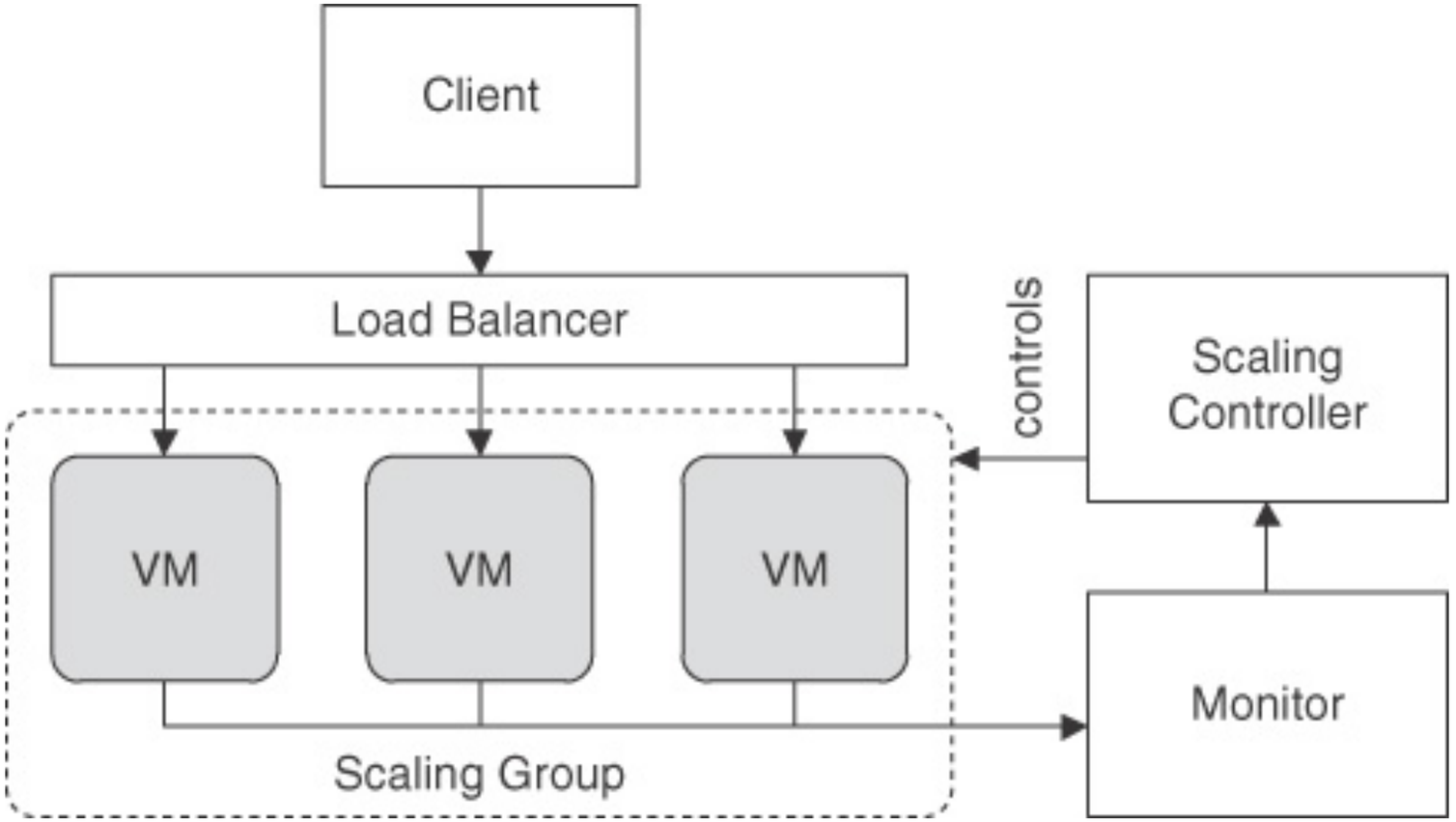


Consistency

- Given the possibility of failure, it is prudent to replicate persistent data
- Consistency is maintained in a distributed system by introducing locks that control the sequence of access to individual data items
- CAP (Consistency, Availability, Partition Tolerance) theorem
  - Eventual consistency
  - NoSQL Databases

Elasticity

- Monitoring used as input to scaling”



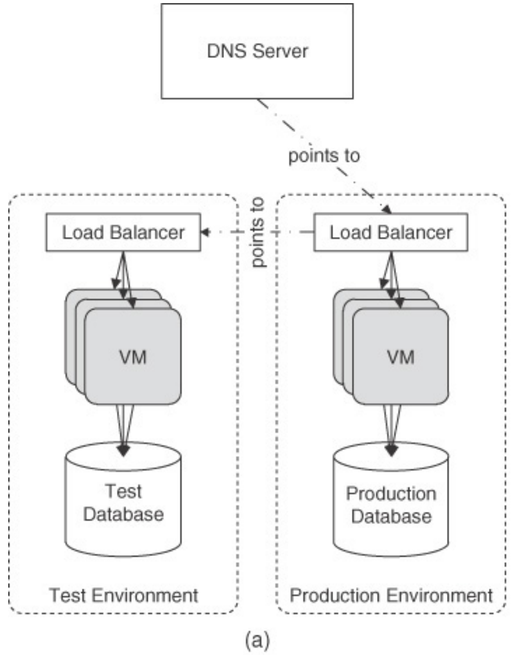
DevOps Consequences of the Unique Cloud Features

Cloud and DevOps

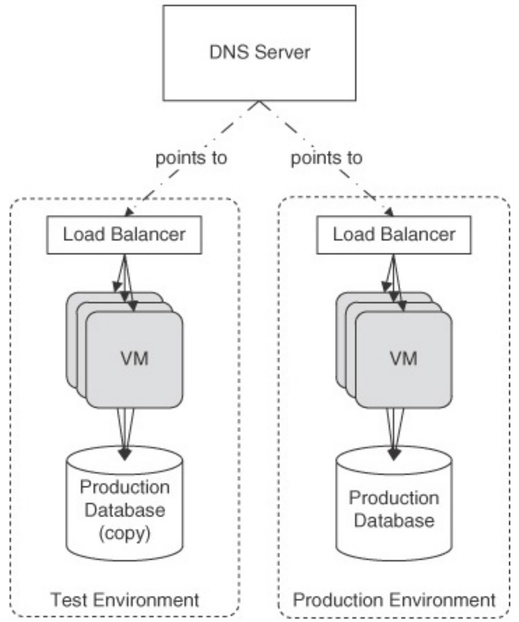
- Three of the unique aspects of the cloud that impact DevOps are:
  - the ability to create and switch environments simply,
  - the ability to create VMs easily, and
  - the management of databases.

Environments

- a set of computing resources sufficient to execute a software system, including all of the supporting software, data sets, network communications, and defined external entities necessary to execute the software system.
  - “self-contained except for explicitly defined external entities”
- Cloud ~> multiple environments during the development, testing, and deployment processes



(a)



(b)

Summary

- The cloud has emerged as a major trend in IT during recent years.
  - Its characteristics include metered usage (pay-per-use) and rapid elasticity, allowing the scaling out of an application to virtually infinite numbers of VMs.

- The cloud rests on a platform that is inherently distributed and exploits virtualization to allow rapid expansion and contraction of the resources available to a given user.

Summary

- IP addresses are the key to accessing the virtualized resources and are associated with URLs through the DNS entries and can be manipulated to allow for the various forms of testing through the isolation of environments.

- Within large distributed environments, failure of the individual components is to be expected. Failure must be accommodated. The accommodations involve management of state and recognizing and recovering from requests that take an exceedingly long time.

- From an operational perspective, controlling the proliferation of VMs, managing different database management systems, and ensuring the environments meet the needs of the development and operations tasks are new considerations associated with the cloud.

For Further Reading

- NIST's definition of the cloud is part of the special publication SP 800-145 [NIST 11].
- The latency numbers for different types of memory and network connections are derived from http://www.eecs.berkeley.edu/~rcs/research/interactive_latency.html
- James Hamilton from Amazon Web Services gives insights into failures that occur at scale in the presentation at http://www.slideshare.net/AmazonWebServices/cpn208-failuresatscale-aws-reinvent-2012
- Memcached system's website can be found at http://memcached.org/
- More information about HDFS and its architecture is available:
  - http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
  - http://itm-vm.shidler.hawaii.edu/HDFS/ArchDocOverview.html
- Netflix's Janitor Monkey helps to keep VM and image sprawl under control; see the following website: https://github.com/Netflix/SimianArmy/wiki/Janitor-Home
- The CAP theorem was first proposed by Erick Brewer and proven by Gilbert and Lynch [Gilbert 02]