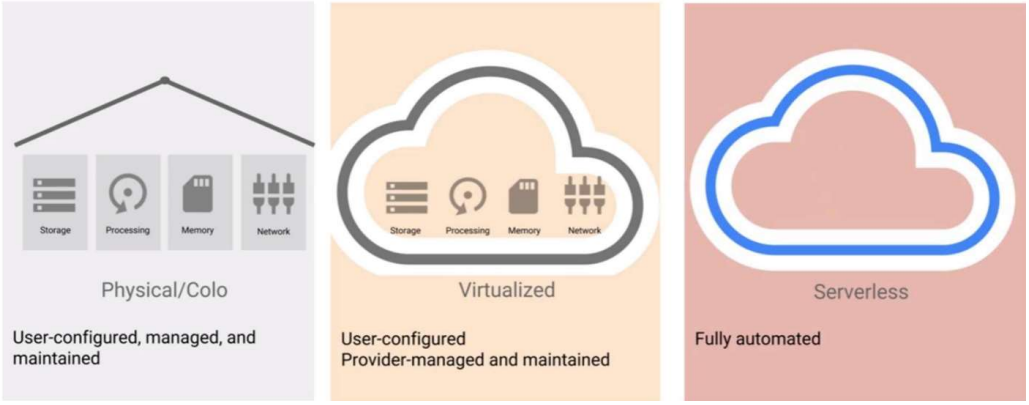
Introduction to Cloud Computing

What is the cloud?

An approach to computing that's about internet scale and connecting to a variety of devices and end points

Cloud Computing as a Utility:

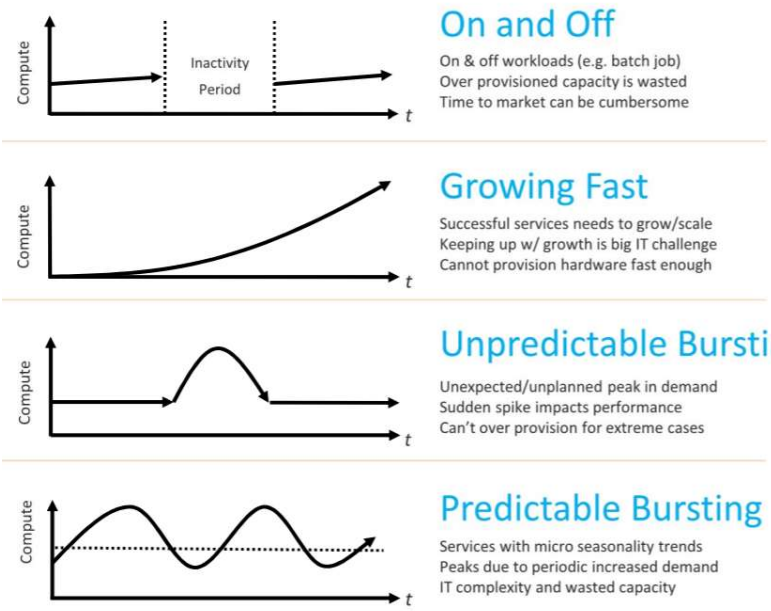
- The idea of cloud computing is to turn computing power into a utility like electricity or water
- You only pay for the computing resource that is deployed on your behalf, which could be billed by the second, minute, or hour
- If you need more compute power you just buy it, you can focus on making the service, not deploying it
- The term 'elastic computing' is used frequently to describe this
- You can automate resource scaling if demand goes up or down



Why Cloud?

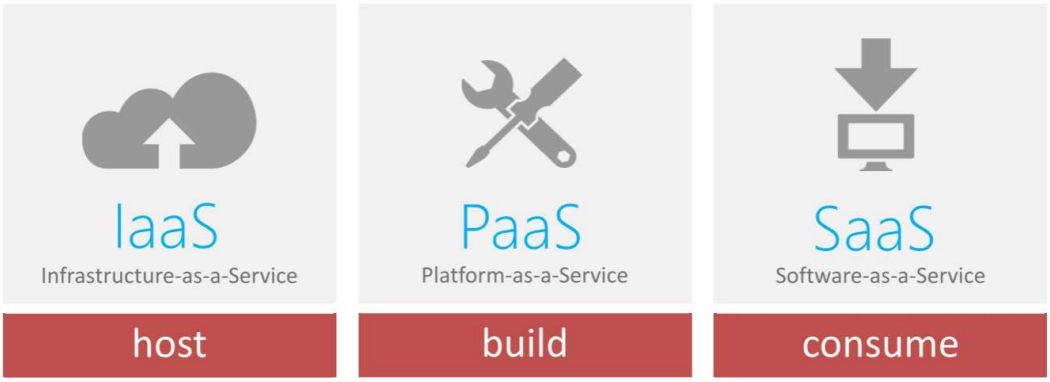
- Rapidly setup environments to drive business priorities
- Scale to meet peak demands, robust redundancy
- Increase daily activities, efficiency and reduced cost

Cloud Computing Patterns

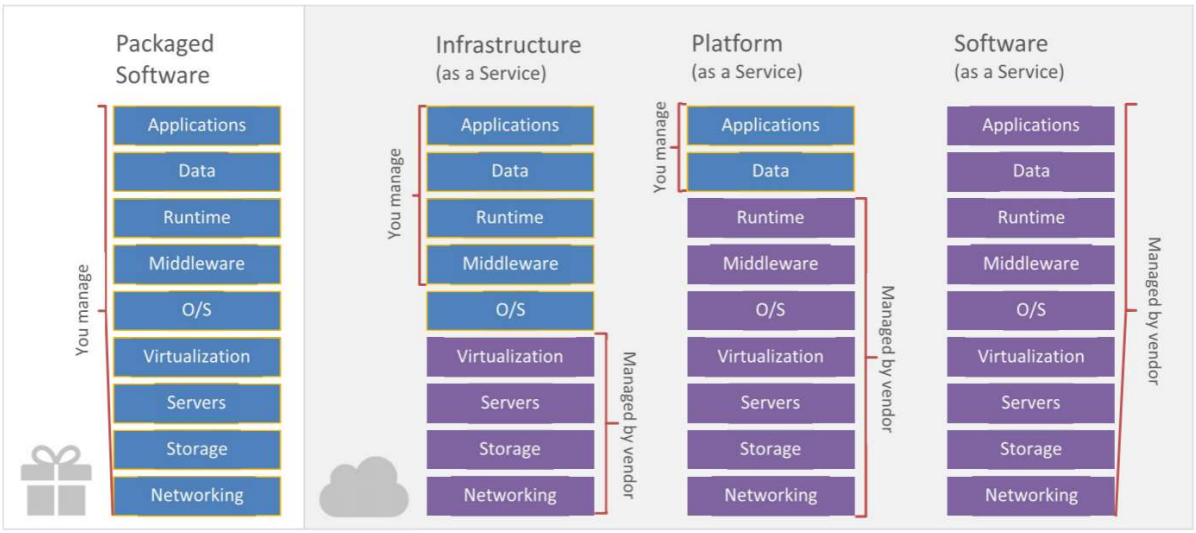


- PaaS: you build it (any language), the cloud takes the code and runs it
- IaaS: interoperability, runs on any cloud provider, or locally
- Automatic OS and Software patching
- Elasticity and scale
- Utility billing approach (only pay for what you use)
- Redundancy – compute and storage, geo-located, etc.
- Focus on ensuring your data is available and secure, without worrying about server infrastructure

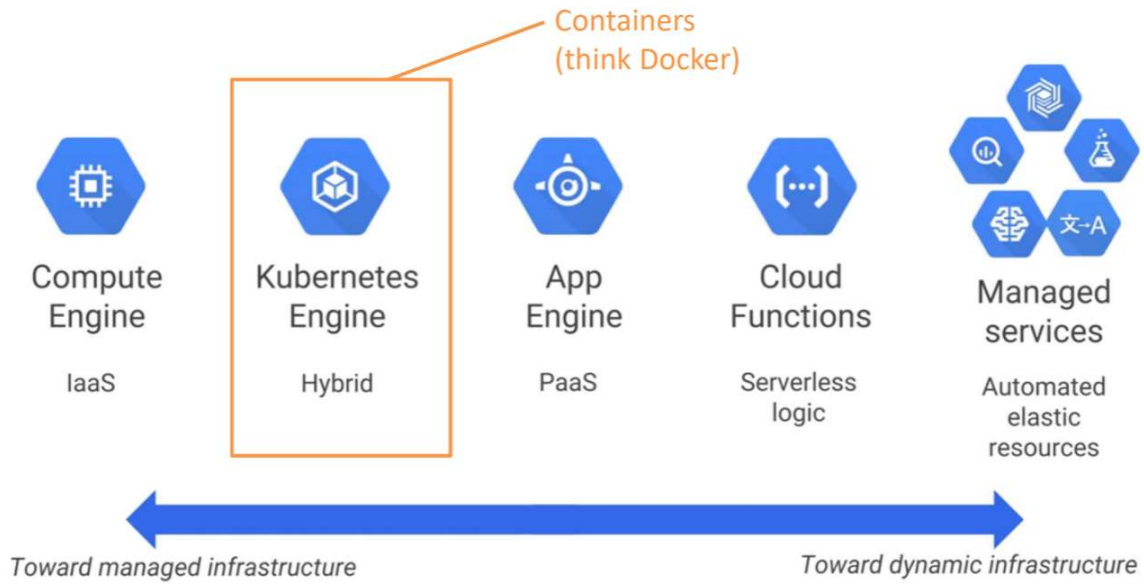
Cloud Computing Service Models



Cloud Computing Stacks



Cloud Computing Stacks – Google's View (applies to all!)



Cloud Benefits:

- Cost of servers and equipment
- Software updates
- Cost of running
- Cost of related services e.g. cooling and UPS
- Reduced downtime
- Cuts operational costs
- Allows the IT Department to focus on strategic projects instead of datacentre
- Remote workers can login anywhere and access apps
- Another company hosts your applications

Where is the Cloud?

Datacentres

- Collection of servers where applications/data are hosted
- A datacentre refers to a single geographical location in which servers are housed
- Hundreds of servers to a room/create – very expensive and very hot
- Rack mounted servers reduces costs and required space, with blade servers and significant virtualisation
- Datacentres are enormous, with security tighter than airports

Cloud Computing Scenarios:

Ideal for Applications needing…

- Scalability
- Availability
- Fault tolerance

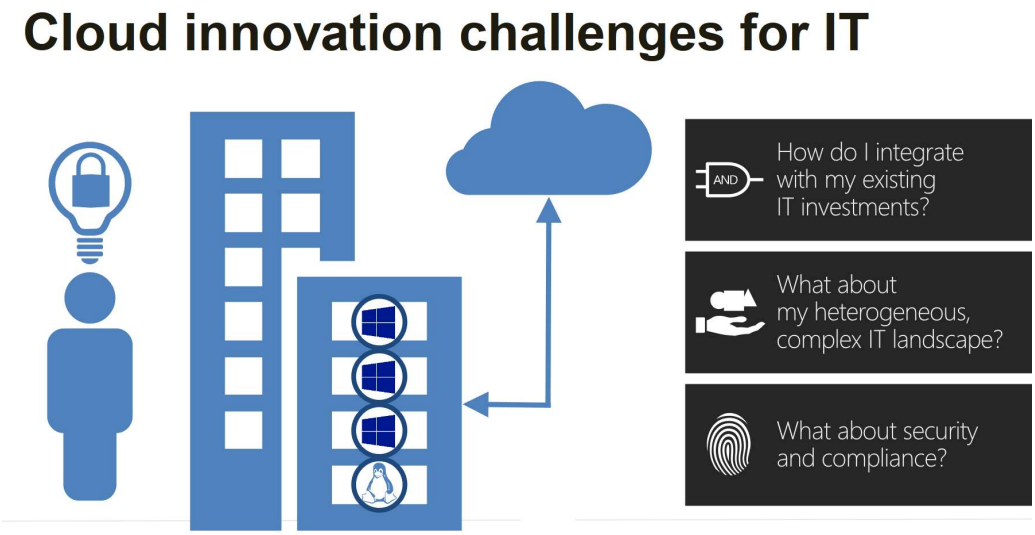
Common application uses…

- Websites
- Compute intensive apps
- Device applications
- Web APIs
- Social games
- Sensing
- Big Data analysis

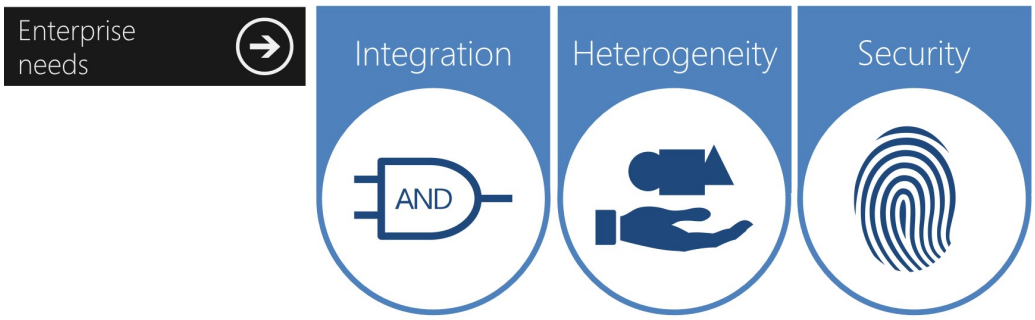
Summary:

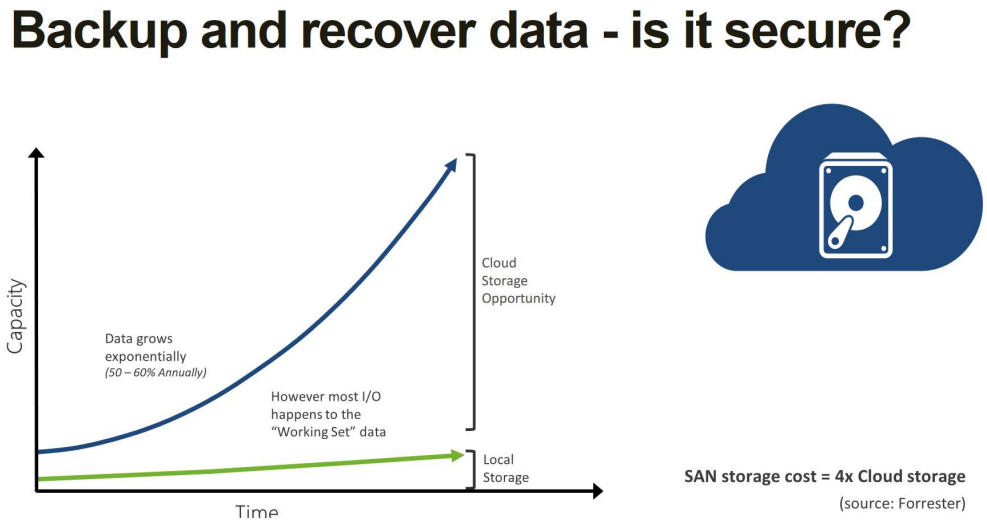
- Cloud computing makes use of the internet to allow a single point of access to services provided by a network of connected servers
- The cloud computing infrastructure distributes tasks and data around a number of large, connected resource centres (datacentres)
- Data and tasks must be managed so that failure of a component will not result in the loss of data
- The aim of cloud computing is to make computing resource into a utility which can be allocated and paid for in the same way as electrical power

PaaS



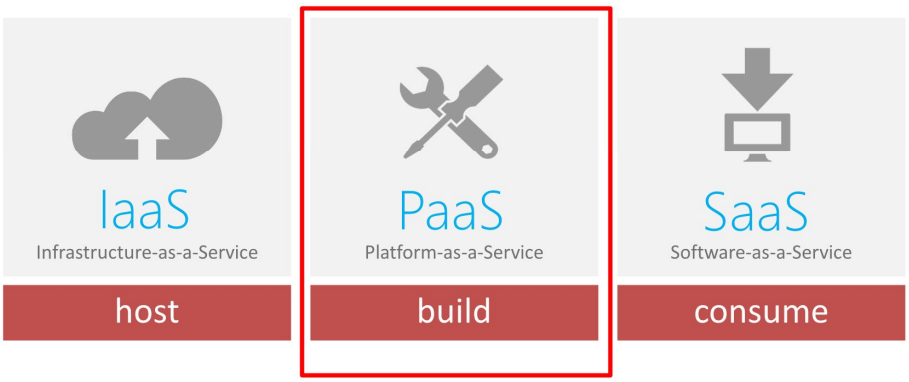
Choosing between PaaS and IaaS



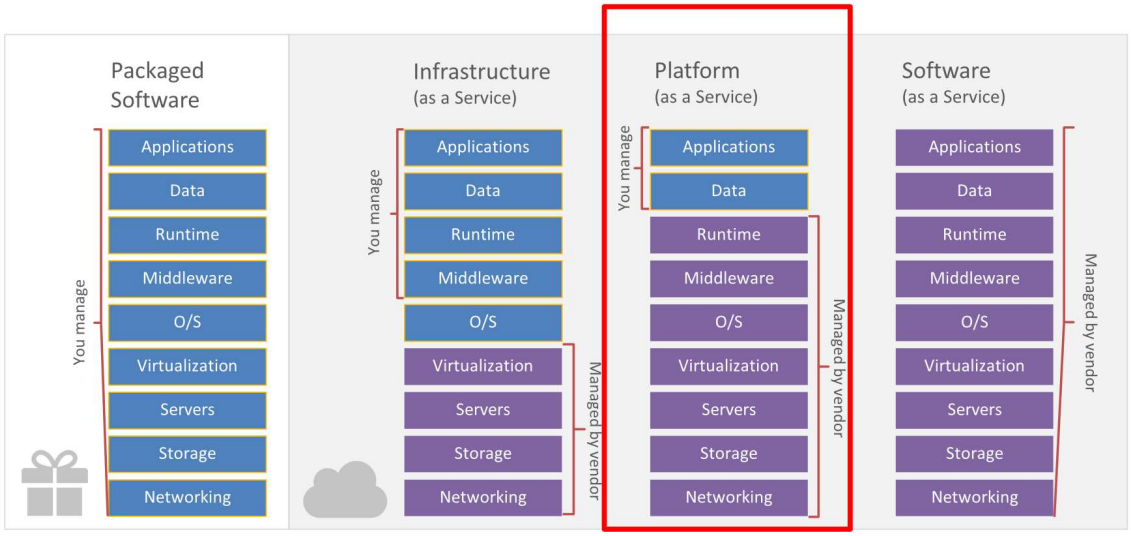


Everything here is easier with PaaS but there are trade-offs with cost and control.

Cloud Computing Service Models



Cloud Computing Stacks - PaaS



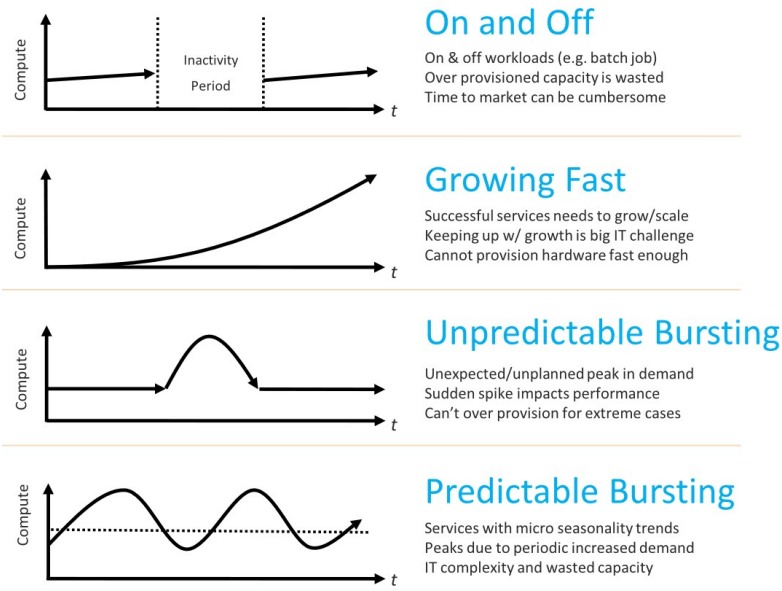
The Benefits of PaaS

PaaS is faster
Reason: There's less work for developers to do
Benefit: Applications can go from idea to availability more quickly

PaaS is cheaper
Reason: There's less admin and management work to do
Benefit: Organisations spend less supporting applications

PaaS is lower risk
Reason: The platform does the heavy lifting, leaving fewer opportunities for error
Benefit: Creating and running applications gets more reliable

Cloud Supports All Patterns

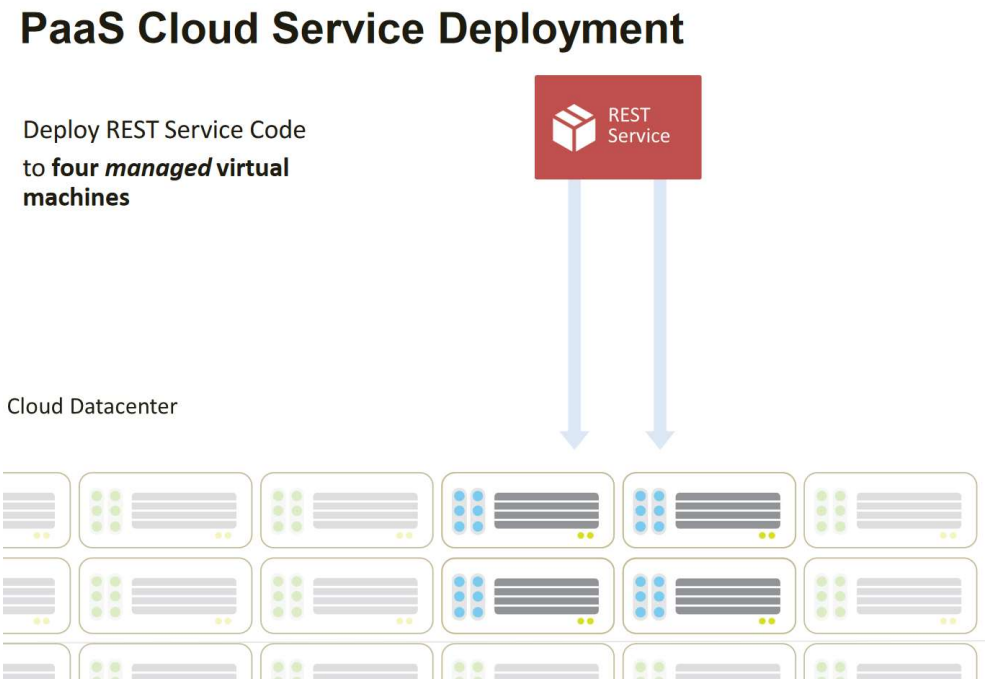


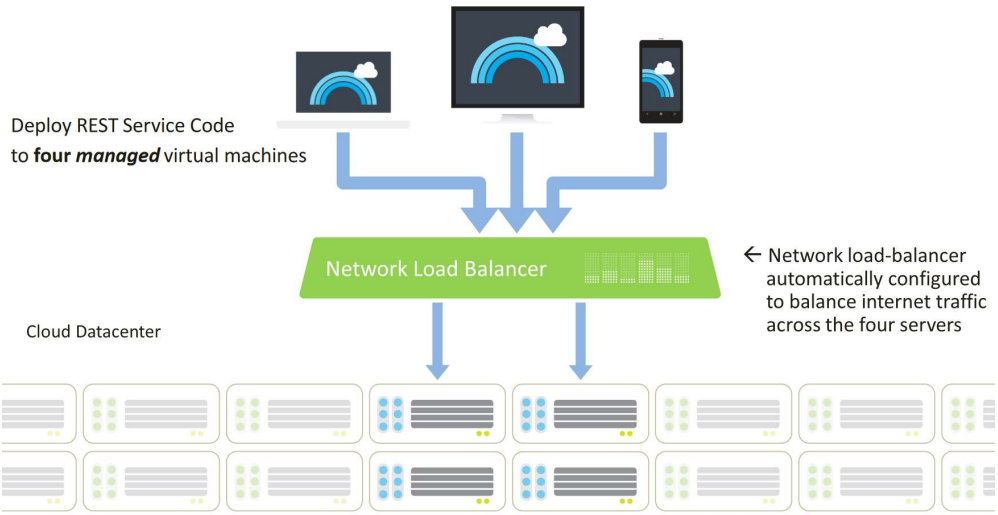
PaaS

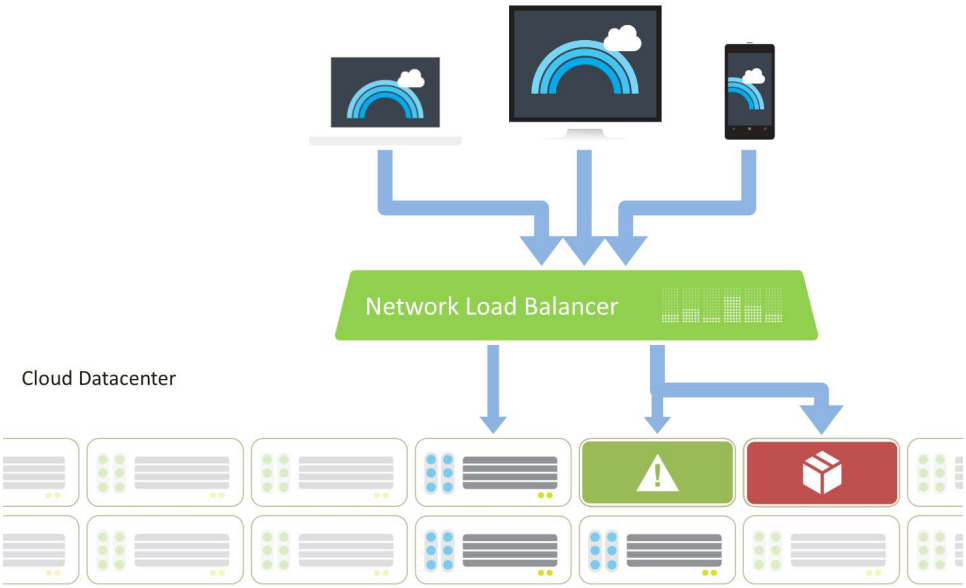
- You build it, PaaS runs it
- Most Cloud Vendors offer PaaS, other examples are website host providers; they provide a software platform to upload your website files and they work with no maintenance from you

PaaS Deployments

- PaaS services are essentially managed – all the configuration is done for you
- PaaS includes deploying code, load balancing, instance recovery
- Other benefits include automatic updates and backups

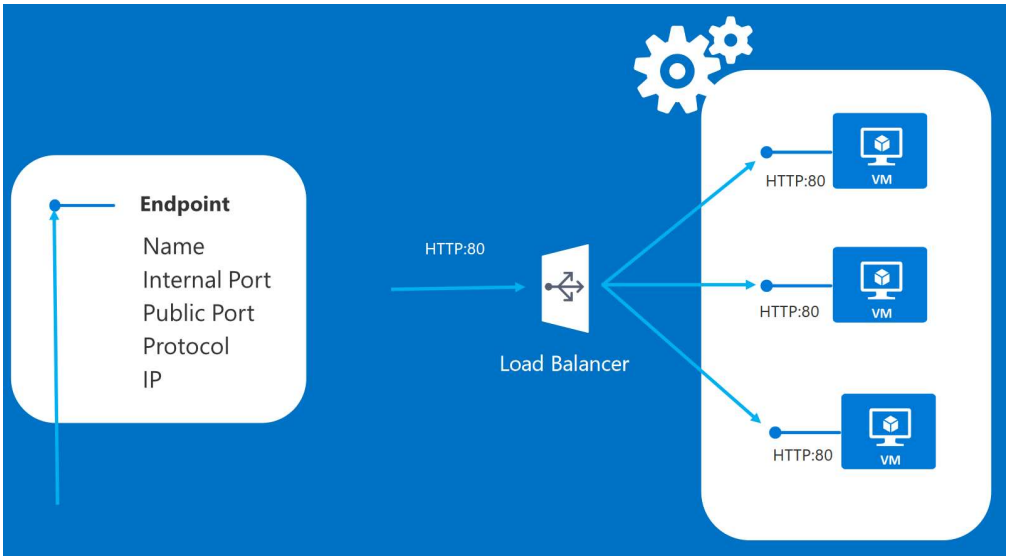






“PaaS watches your aaS”

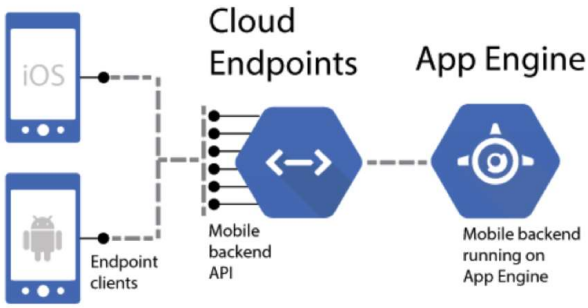
PaaS Deployment Example:



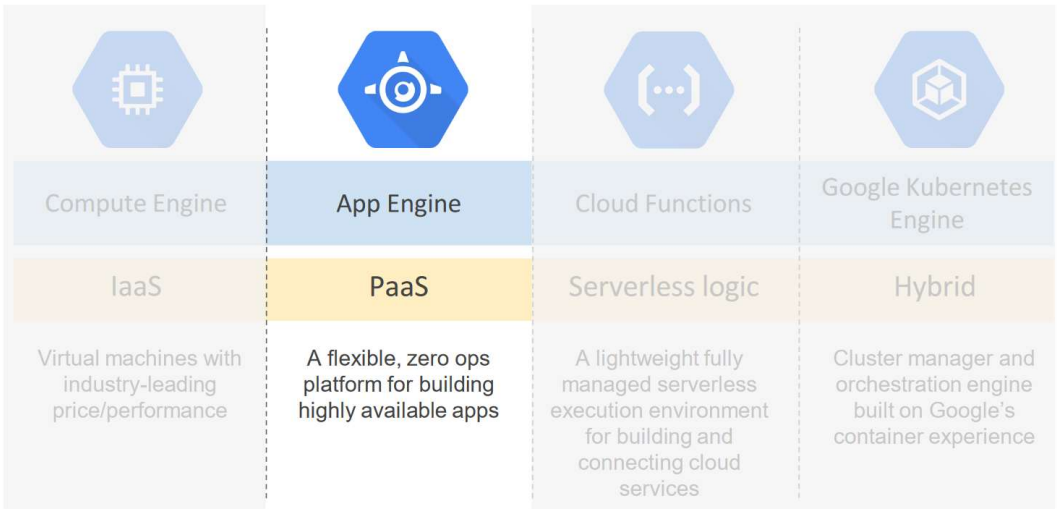
- A cloud PaaS service can have multiple compute instances running inside it, this one has 3 VMs
- Essentially, it is a fully-managed PaaS service with 3 running VMs
- The same PaaS services was also automatically load balanced across the 3 virtual machines
- All traffic reaches the same public internet endpoint, the load-balancer then distributes requests to the 3 instances dependent upon their load at the time
- With PaaS, everything is fully managed for you, but you lose fine grain control as a result
- Unlike IaaS where you have to do much more of the configuration, but gain more control of the infrastructure

PaaS Deployment- Google App Engine

- On the right a cloud PaaS service is hosting a backend service for a mobile app
- It is running on Google's PaaS App Engine service
- App Engine will automatically scale the service to handle traffic
- Scale means up and down!
- PaaS services really are 'elastic'

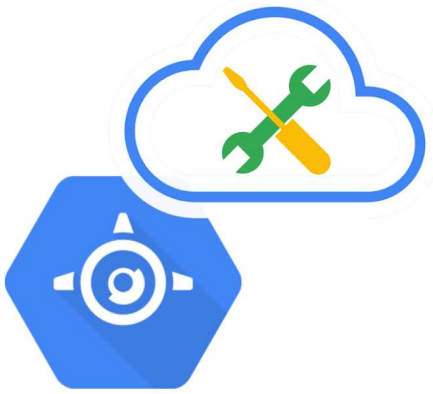


Google App Engine



App Engine - platform-centric solution

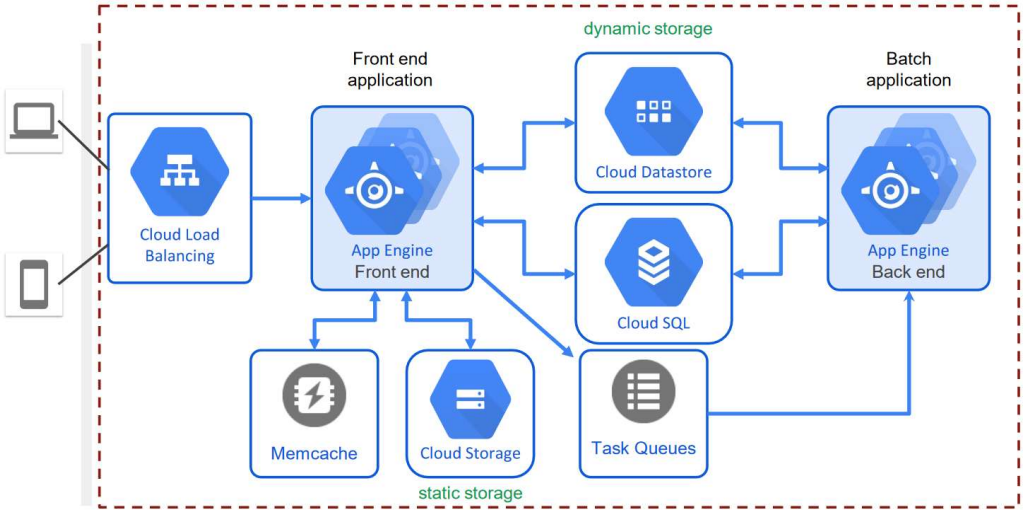
- Type of PaaS
- No need to buy, build, or operate hardware/infrastructure
- No managing servers or configuring deployments
- Focus on app development instead of operations
- Use a range of languages and tools



App Engine - two environments

Standard environment	Flexible environment
Fully-managed	Docker container support
Scale to zero	VMs exposed
Specific versions of supported languages	Any language in your container
Changes/configuration limited	More options for infrastructure customization and configuration for performance

Example of App Engine Architecture



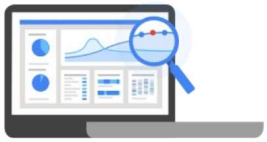
Addresses key needs of developers

Multiple storage options

Automatic scaling

Load balancing

App versioning



Monitoring and logging

Security

PaaS Deployment Summary:

- The basic concept is the same across cloud platforms – PaaS services will run in a fully managed cloud virtual machine instance with runtime code packaged inside
- Fully managed virtual machines are not generally accessible in terms of ‘remoting’ into the underlying OS, hence the name fully managed (and auto-updated/patched)
- You generally configure PaaS service through a GUI/Shell then deploy, i.e. you can simply switch features on and off through the cloud dashboard, or shell commands
- PaaS is faster to develop for, but you surrender fine-grain control with far less customisation and interoperability

Cloud Infrastructure-as-a-Service (IaaS)

PaaS Revisited:

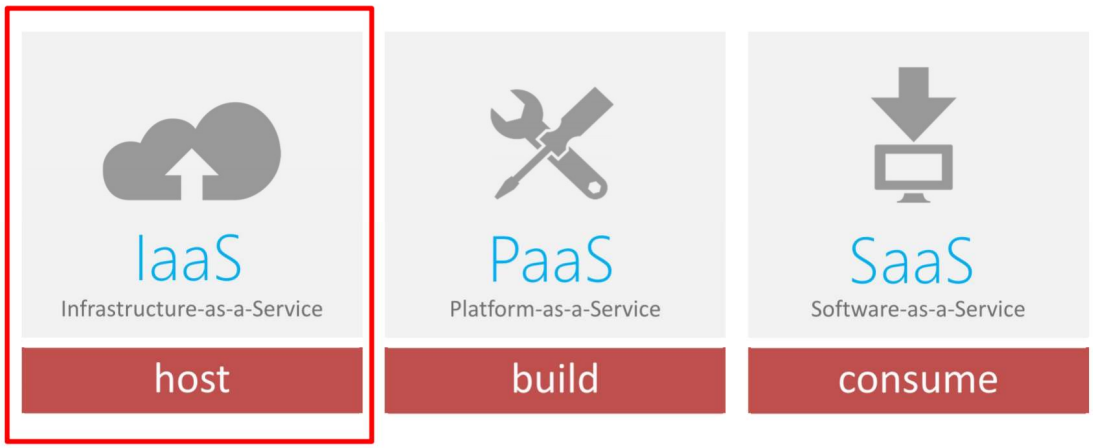
The simplicity of PaaS is that you can give it your code and it will just run…

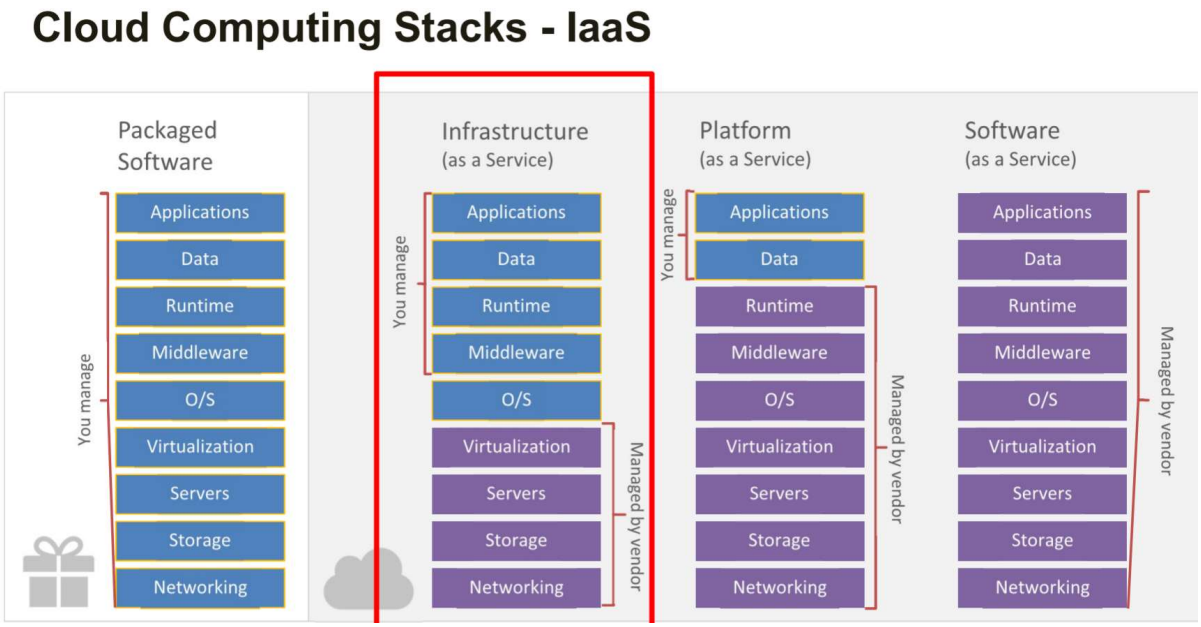
- Minimal configuration
- No OS installation/framework sdk installation
- No configuration/patching/maintenance

Developers can simply concentrate on writing code instead of grappling with configuration issues

PaaS or IaaS?

If a business needs more control over their cloud infrastructure and wants to migrate its on-premise virtual machines, IaaS is the choice





- Regardless of whether you are using compute through PaaS or IaaS, your cloud service will be running on a virtual machine
- For PaaS, the virtual machine is abstracted away and you just manage the data and application code, hence it is called a managed service
- For IaaS, you manage the virtual machine, and with that have full access to the VM's OS and responsibility for updates etc.
- Terminology: a single virtual machine is known as an **instance**
- An instance can be configured, e.g. number of CPU cores, RAM, disk size, etc.

Virtualisation:

- Virtualisation has been around for some time
- In the 70s, IBM pioneered hardware-level virtualisation
- The key benefit back then, as it is still now, was to run multiple operating systems on the one physical machine
- Nowadays virtualisation of services helps to prevent under-utilisation of traditional on-premise compute and datacentre resources
    - Also allows rapid testing and development of specific server images
    - Supports business continuity by encapsulating entire systems into single image files – e.g. virtual hard disk (VHD)
    - These can be restored anywhere (i.e. to different cloud vendors)
- A virtual machine (VM) is an operating system (OS) or application environment that is installed on software which imitates dedicated hardware. The end-user has almost the same experience on a virtual machine as they would on dedicated hardware.

Virtualisation – Hardware Acceleration:

- Hardware acceleration features for VMs are now ubiquitous (present, appearing, or found everywhere)
- Provide performance enhancements by allowing direct access to hardware
- Intel's Virtualisation (VT-x) is built into its newer x86 CPUs and provides specific x86 CPU hardware instructions to manage VMs faster and without this, all VM management is done purely through software which is slower
- Intel's Graphics Virtualisation Technology (GV-T) was recently introduced as Intel Iris Pro and allows the on-die GPU to be dedicated to one or more VMs. Physical machines can also have a discrete GPU

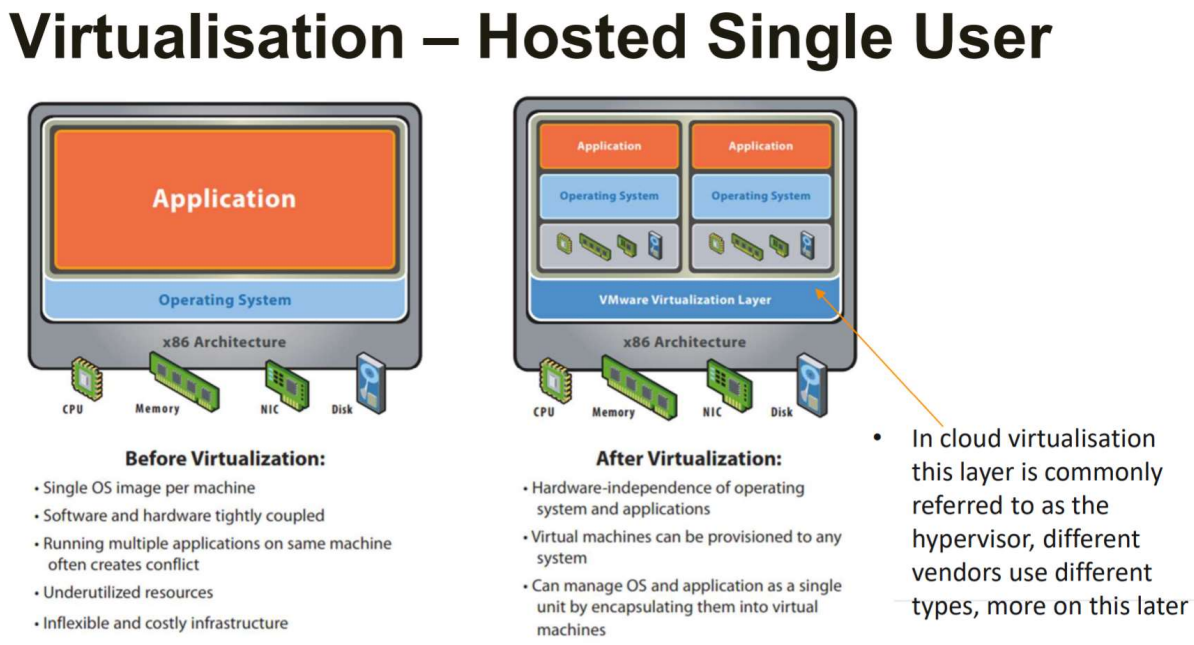
Virtualisation Approaches:

- There are two main approaches to virtualisation in x86 systems:
    - Hosted architecture
    - Hypervisor (bare-metal) architecture
- Hosted has better compatibility across different types of hardware, it is installed on top of the host OS
- Hypervisor has great scalability, reliability, and performance. It is installed directly over the hardware and device drivers

- Hosted is generally directed at the individual end-user and usually comprised of low cost or ‘free’ software i.e. Virtual Box, VMware player
- Hypervisor is the approach used by cloud vendors for their Cloud OS

Virtualisation – Hosted Architecture:

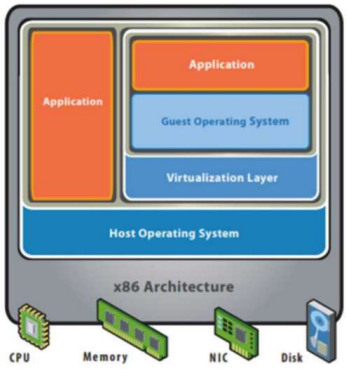
- Hosted virtualisation runs ‘on top’ of the host OS as a guest OS
- Guest OS is abstracted away using VM software such as Virtual Box or VMware workstation
- Type of approach used to install personal or small group Windows on Linux and vice-versa
- Does not normally have direct access to physical hardware
- Access to hardware features varies greatly between VM software vendors, and many depend on CPU hardware virtualisation features

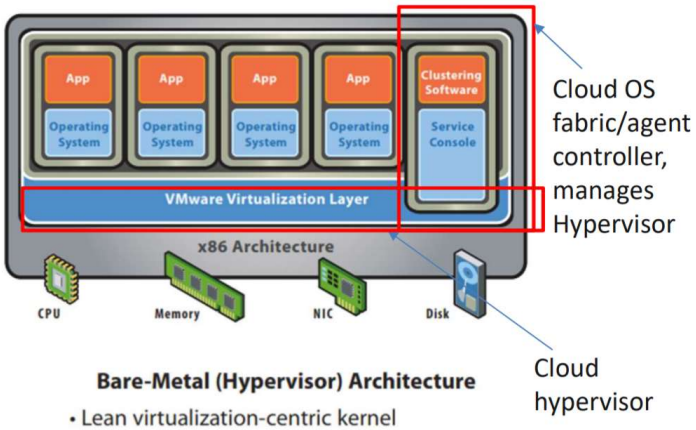


Hypervisor (Bare-Metal) Architecture:

- The virtualisation software (hypervisor) is installed on the bare-metal physical machine first (Server Blade) before any guest OS
- Sits on top of hardware device drivers
- Direct access to hardware resources
- Each guest OS runs a process (a VM instance)
- Scales with great performance
- hypervisors of course used in the cloud extensively
    - Amazon’s Xen Hypervisor
    - Azure Hyper-V Hypervisor
    - VMware ESX Hypervisor

Cloud Bare-Metal Hypervisor



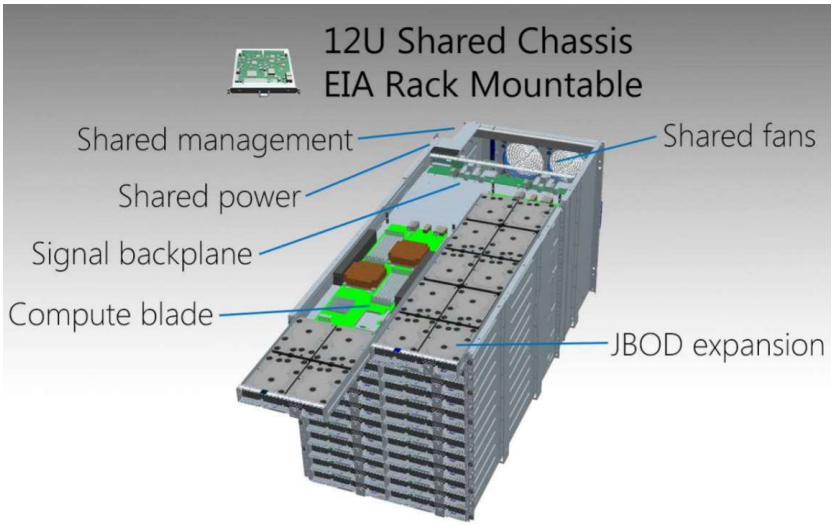


Hosted Architecture

- Installs and runs as an application
- Relies on host OS for device support and physical resource management

Bare-Metal (Hypervisor) Architecture

- Lean virtualization-centric kernel
- Service Console for agents and helper applications

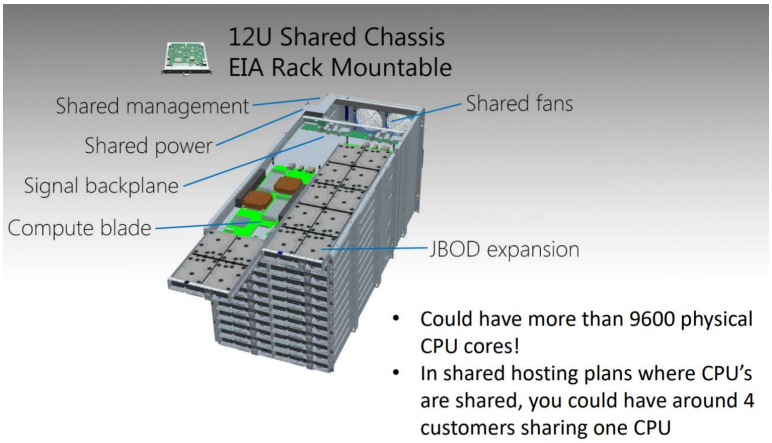


Cloud Virtualisation

- 24 Server 'Blades' in a rack
- Each blade holds 20 physical CPUs with up to 10 cores per CPU = 200 CPU cores per blade
- A single rack can hold 24 blades: 24 x 200 = 4800 cores in a rack! Potentially 4800 VMs
- Modern Xeon CPUs can have as much as 20 cores per CPU = 9600 CPU cores per rack.

Microsoft cloud server specification *Compute blade*

Processor	
CPU	Dual Intel® Xeon® E5-2400 v2 family
Core QTY	Up to 10 cores / CPU, 20 / Blade
TDP Wattage	Up to 95W
Memory	
Memory Busses and DIMM Slots	3X memory bus / CPU, 6 / Blade 6 DIMM slots / CPU, 12 / Blade
DIMM Type / Speed	16GB, 2Rx4, 1333MHz, 1.35V
Max Capacity	192 GB / Blade
On-Board Devices	
Storage Controller	Intel® C602 PCH
SATA	4 ports @ 3.0 Gb/s
SATA	2 ports @ 6.0 Gb/s
Server Management	
Chipset	BMC-lite serial thru Chassis Manager
Interface	REST API, CLI thru Chassis Manager
System Firmware	
Version, Vendor	UEFI 2.3.1, AMI
Security	TPM 1.2, Secure Boot
Blade I/O	
PCI-Express Slots	One Gen3 X16 Riser
Networking	Single or Dual 10G Mezzanine Card
SAS	Dual 4X SAS @ 6G Mezzanine Card



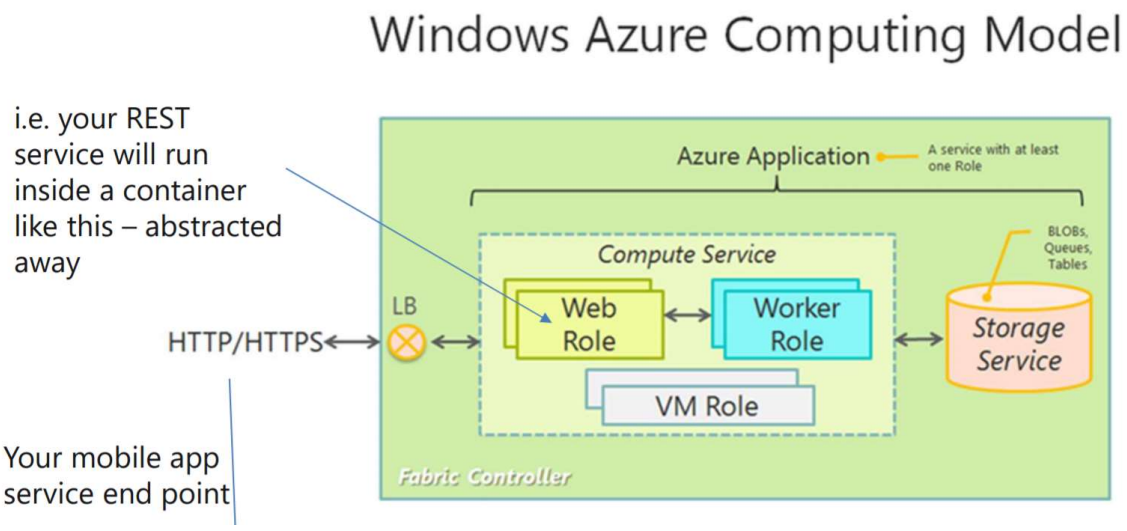
Virtualisation Summary:

- Virtualisation of resources primarily reduces cost
- Also provides optimal performance and utilisation of resources
    - E.g. run 9600 VMs or more on one physical rack
    - Maximise physical resource utilisation
- Two main approaches to virtualisation:
    - Hosted architecture
    - Hypervisor (bare-metal) architecture
- Provides business with optimal solutions for…
    - Server consolidation
    - Costs
    - Rapid testing and development
    - Business continuity

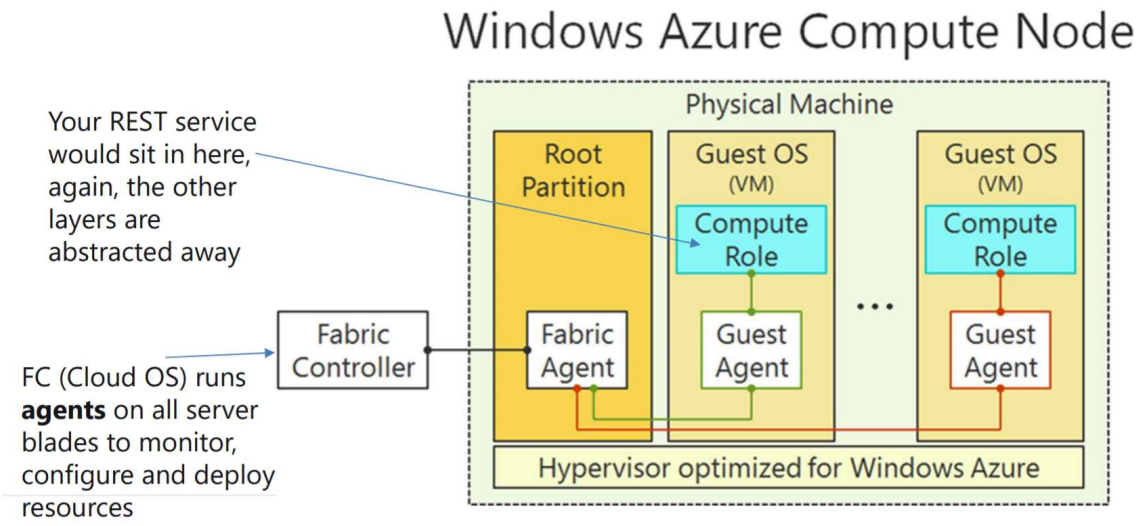
Cloud Operating System – Azure:

- It is important to note that a cloud platform is not just running on a cluster of virtualised Microsoft Server/Linux OS machines
- Cloud providers (i.e. Amazon, Azure, Google) will have their own Cloud OS 'agents' that manage their infrastructure across many services
- For Azure/Google it is called the Fabric Controller (FC) – where fabric is an abstraction of resources (PaaS Services, VMs, etc.) that are dynamically created on demand via a Hypervisor service – the hypervisor is an agent/service of the FC
- All cloud compute, data, and networking resources are managed by the FC
- For example when you deploy a VM on Azure, the FC takes care of this and does all required configuration and deployment to the server rack

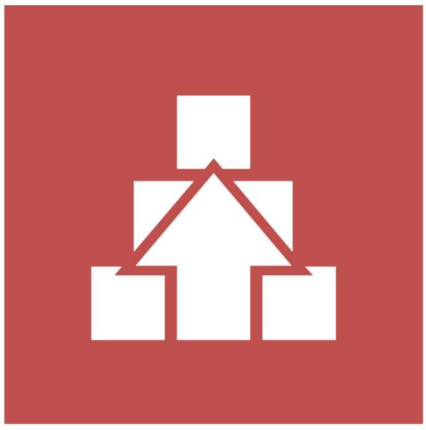
Azure Cloud OS – Zoomed In



Cloud OS – Zoomed Out



Azure VM Fault and Update Domains



Fault Domains

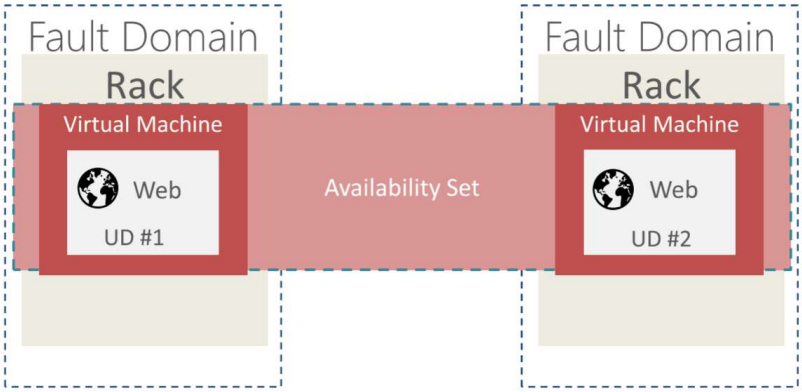
- Represent groups of resources anticipated to fail together
- i.e. Same rack, same server
- Azure Fabric Controller spreads instances across at least 2 fault domains (or two distinct racks)

Update Domains

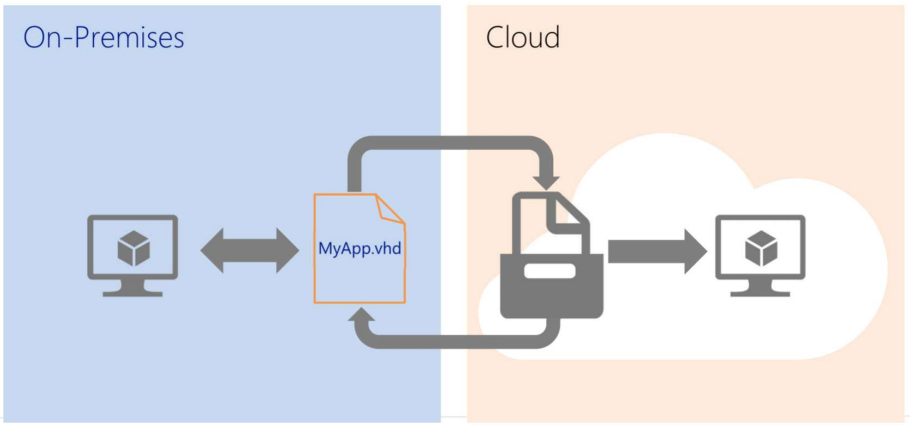
- Represents groups of resources that will be updated together
- Host OS updates honour service update domains

Fabric Controller spreads role instances across Update Domains and Fault Domains

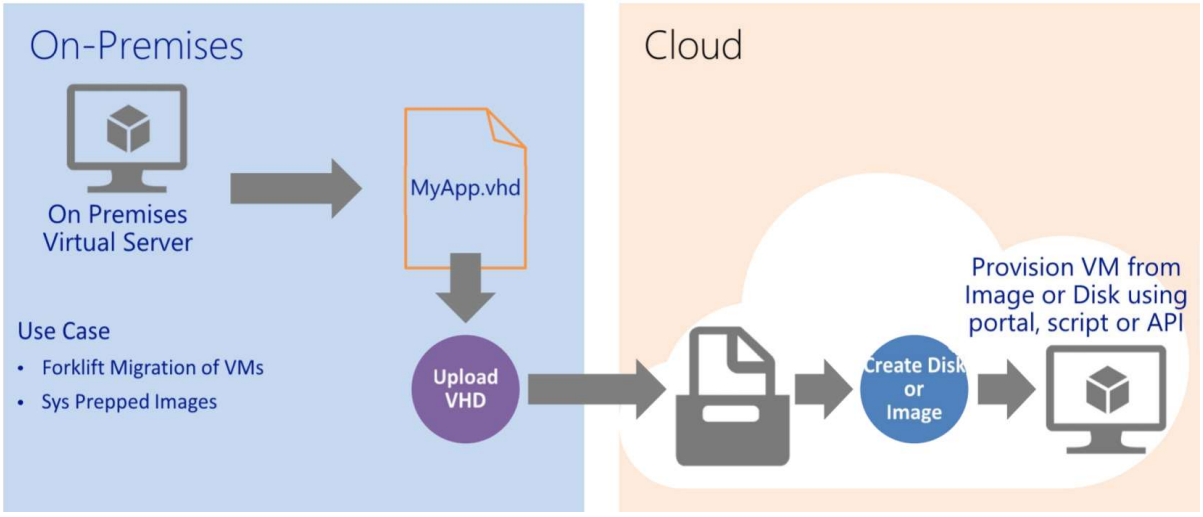
VM Availability Sets



Azure VM Image Mobility

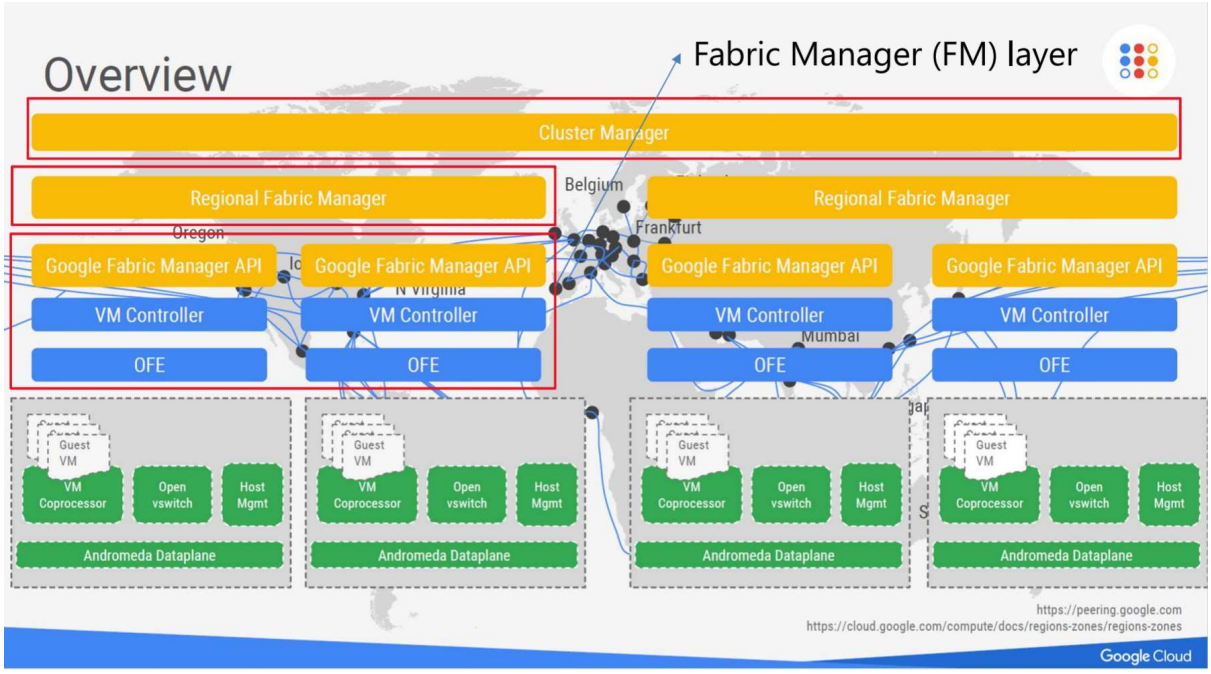


Azure - Bring Your Own VM/VHD



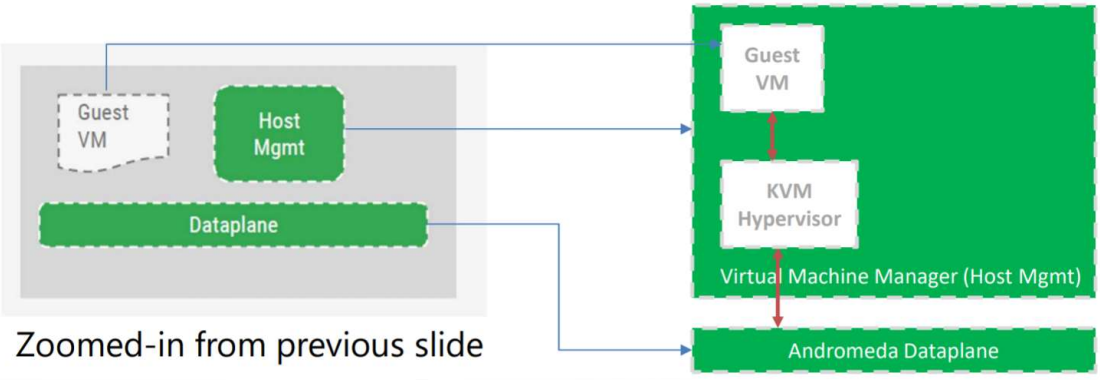
Cloud OS – Google Cloud Platform:

- At the top level of the GCP infrastructure the cloud OS consists of a Global Cluster Manager, which manages…
    - Regional Fabric Manager (specific geographic region)
    - Fabric manager API (similar to Azure)
    - VM controller
    - KVM Hypervisor
- Similar to Azure, when you deploy a VM the request is handled down the service hierarchy until the hypervisor service processes the request, and deploys your VM/service to a specific region
- Difficult to find further technical information – IP issues



Google Cloud – KVM Hypervisor

- After some digging around I was able to create a draft of Google's KVM hypervisor implementation:



Networking

Overview:

- Most cloud services are virtualised resources
- These virtualised resources are networked
- They are physically networked at a lower level in the cloud datacentre; through cables, routing, switches etc.
- Network management is available through the Google Cloud Console

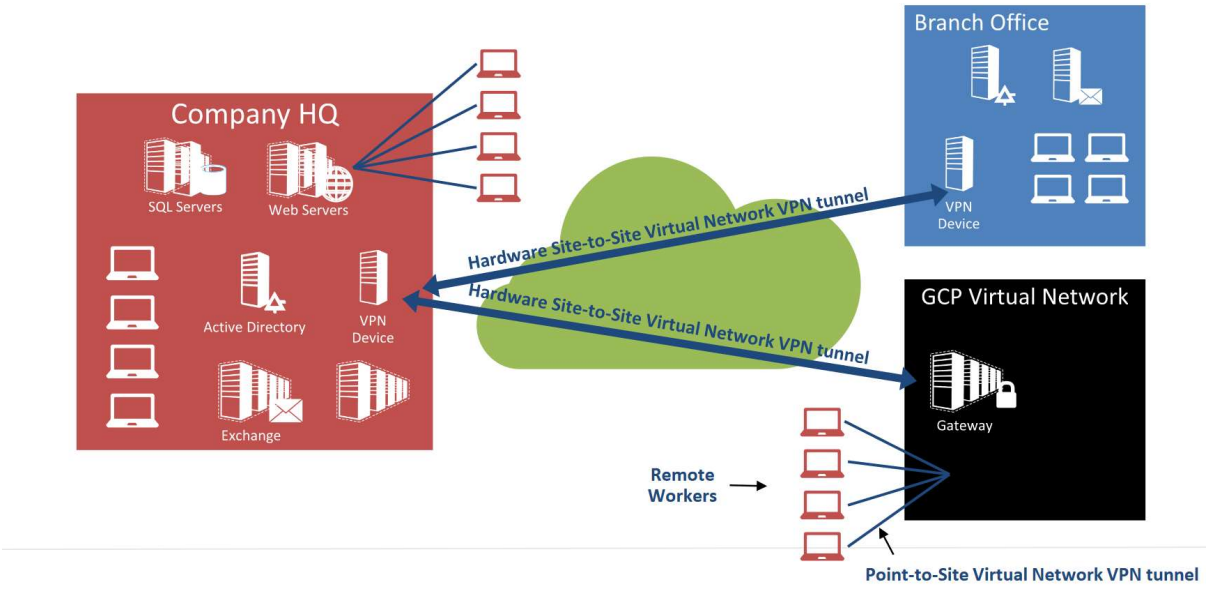
Networks:

- Cloud networking is not configurable by end-users at the physical layer, it is done through a GUI, CLI or deployment manager files
- For Google Cloud Platform, it is called a Virtual Private Cloud (VPC) Network
- End-users can create, configure, and deploy logical networks in the form of SDNs (Software Defined Networks) that utilise the underlying network hardware in the datacentre – virtual networks
- Virtual networks provide secure connections between cloud resources
- They can map over practically any on-premise network to an SDN equivalent
- A business can organise their ICT resources using virtual networks…
    - Create a virtual network that securely connects all company services in the cloud i.e. SQL databases grouped together
    - Create a virtual network between on-premise resources and cloud resources – hybrid cloud
    - Create a virtual private network (VPN) that allows personal computers to securely connect to on-premise company resources, or in the cloud

Consider:

- Four Virtual Machines are running on a server rack
- They all share a single physical **Network Interface Card** (NIC) – aka a network ethernet port
- Each VM will be allocated a unique Virtual IP (VIP) running through the NIC, eg. 192.168.0.2. 192.168.0.3, etc.
- Configured using a virtual SDN
- Allows network traffic specific to each VM to tunnel through the single physical NIC

Virtual Private Network Overview



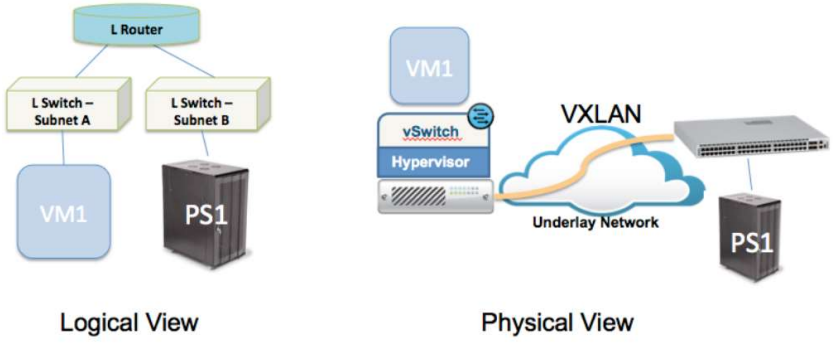
Virtual Networks:

- Services on Google's VPC networks are assigned external (public) and internal (private) IP addresses
- Using these IPs you can create and configure cloud features such as load balancing…
    - Load balancing distributes client traffic evenly over multiple virtual machine instances of your cloud service
    - Can be done locally (internal or external) or geographically
- When creating IaaS resources (Virtual Machines) or PaaS resources (cloud services) you can allocate them, logically, to a VPC at creation

Logical vs Physical:

- Logical networks are how devices appear to be connected, these are SDNs
- Physical networks are how they are actually physically connected via cabling and switches

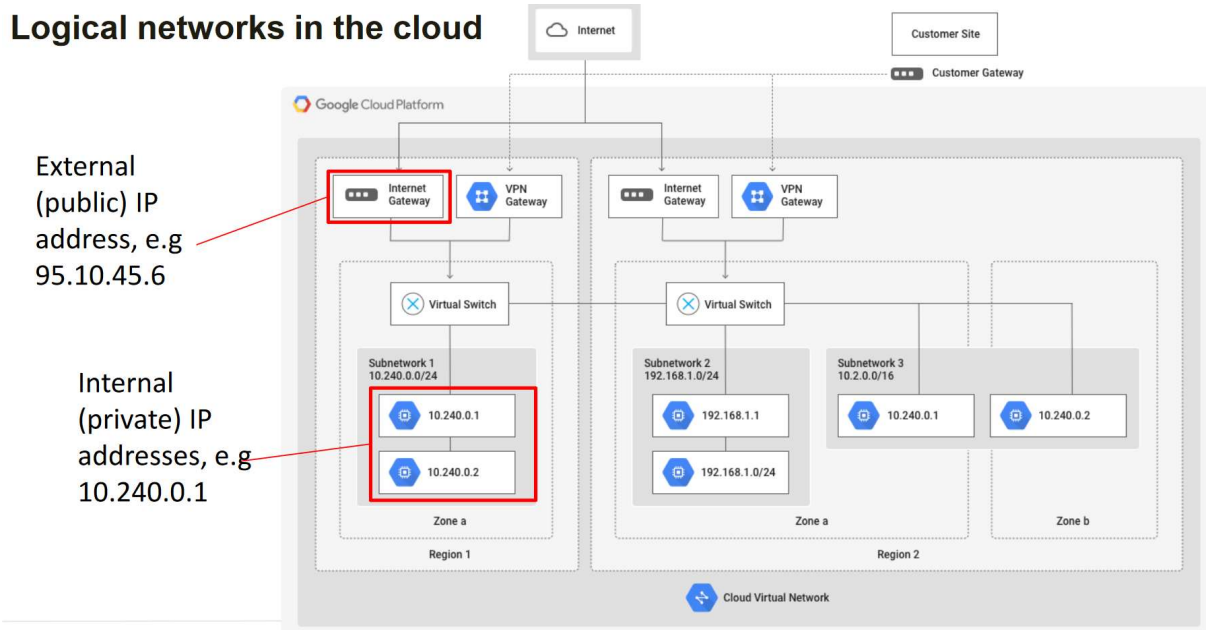
Distributed Logical Routing



Logical View

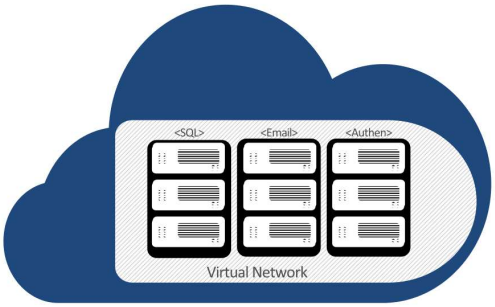
Physical View

Logical networks in the cloud



Virtual Networks – Cloud Only:

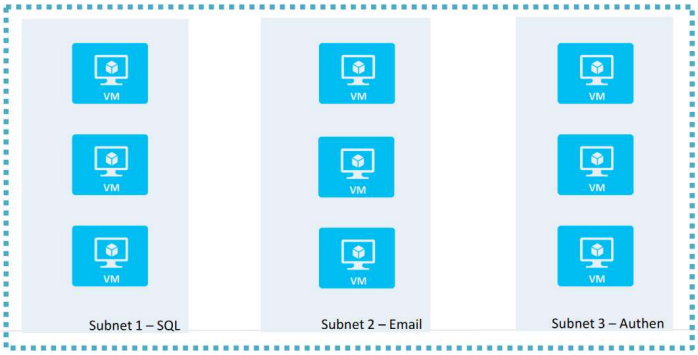
- Logical isolation with full control over network
- Creates subnets, use your internal private IP addresses
- Stable and persistent private IP addresses
- Ideal for large scale cloud-only deployments
- Connectivity options



Virtual Networks - Subnets

- Assigns a specific private IP address space:
  - Subnet 1 (SQL servers): 192.168.10.1 – 192.168.10.15
  - Subnet 2 (Email servers): 192.168.11.1 – 192.168.11.15
  - Subnet 3 (Authentication servers): 192.168.12.1 – 192.168.11.15

Subnets



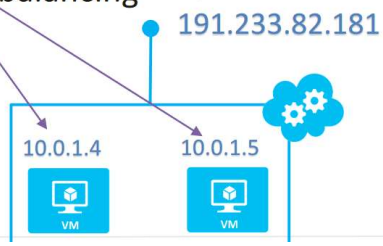
Addressing VMs on a virtual network

External IP

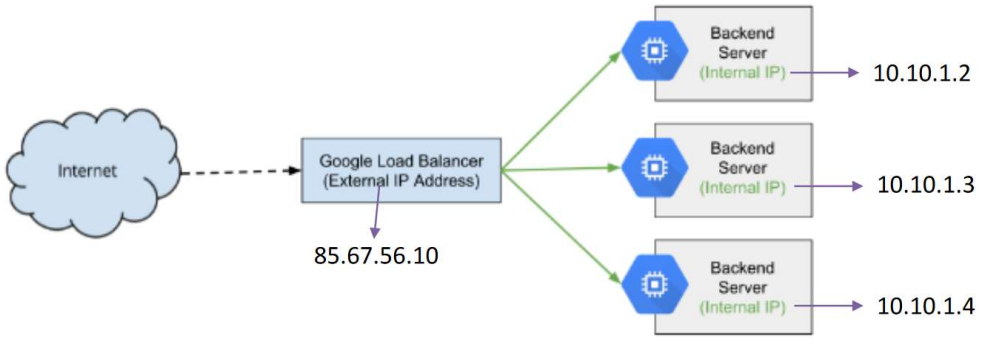
- Public IP address assigned to the virtual machines

Internal IP

- Private IP addresess assigned to the virtual machines from the Virtual Network, also known as the Virtual Private Cloud (VPC) network on the Google Cloud Platform, also used for internal load balancing



Addressing VMs on a virtual network



Virtual Networks Summary:

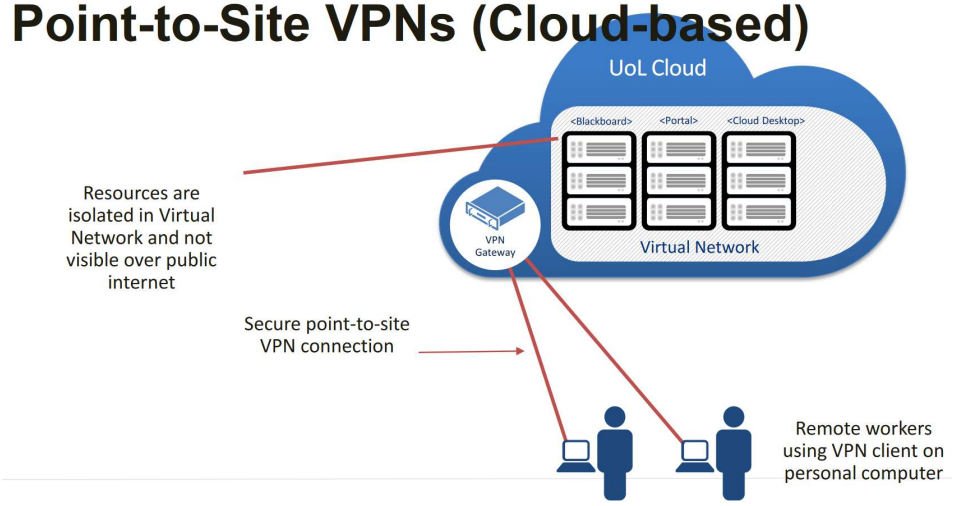
- A virtual network is a logical network overlay over the top of the physical networking infrastructure, generally known as an SDN (VPC network on GCP)
- Resources on the same virtual network, i.e. VMs and cloud services, can securely access each other
- Provides an island of resources that cannot be seen form the internet without a secure connection (VPN)
- Different types:
    - Cloud only (Virtual network completely contained in the cloud)
    - Site-to-site (connects on-premise networks to cloud VPC)
    - Point-to-site (connects client to cloud or on-premise network)

Point-to-Site VPNs:

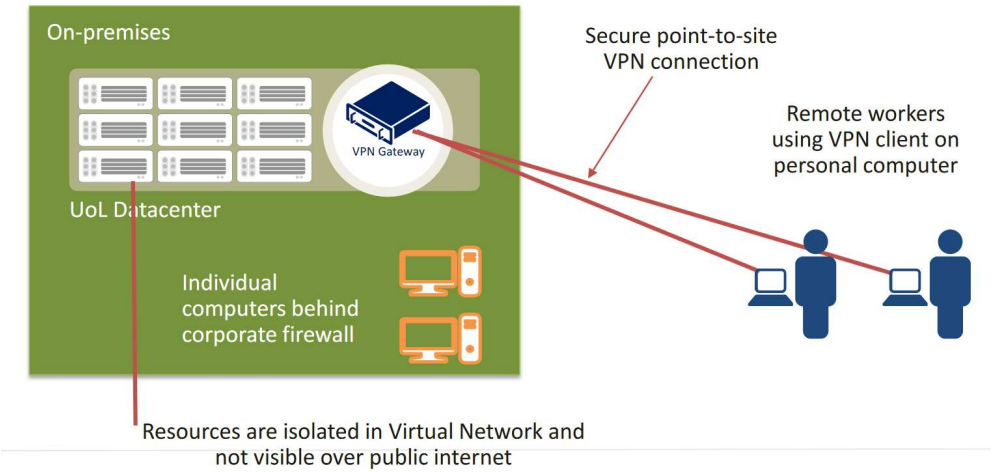
- These can be classed as cloud only networks if the resources are all hosted on cloud
- Could be hosted in on-premise datacentre

- Typically involve remote workers connecting securely to business resources either on the cloud or on-premise
- Provides a securely encrypted ‘tunnel’ from the remote client PC to the resources
- Usually require VPN client software installed
- Windows 7 and onwards have a built-in VPN client

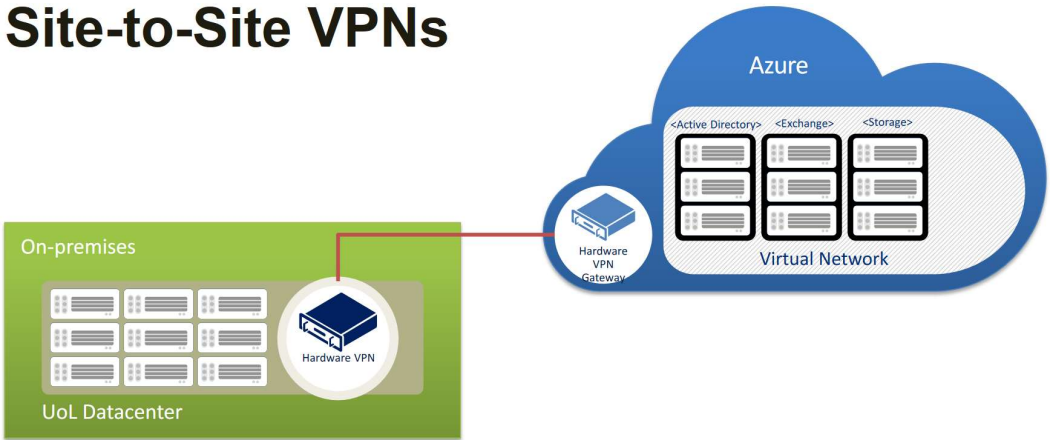
Point-to-Site VPNs (Cloud-based)



Point-to-Site VPNs (On-premises)

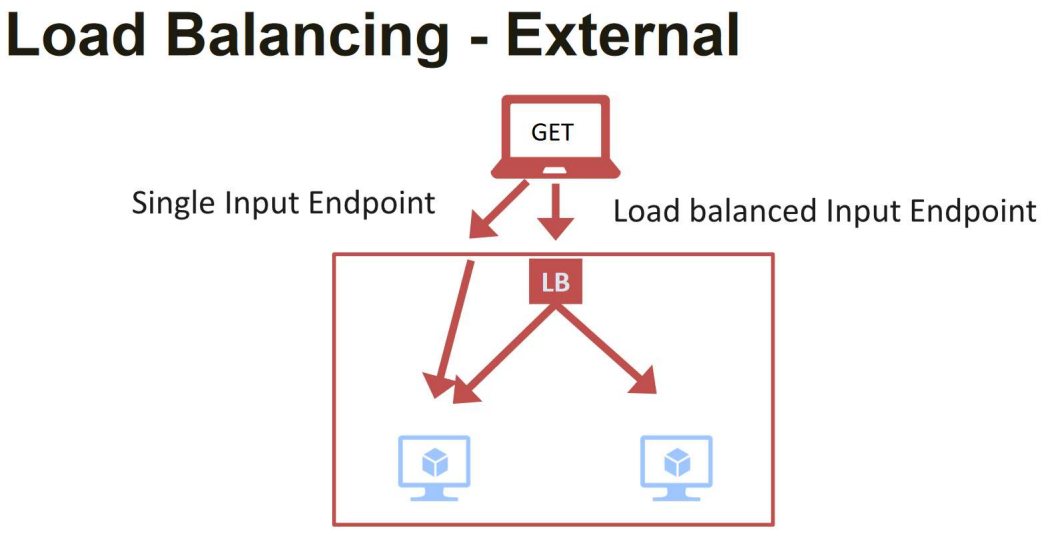


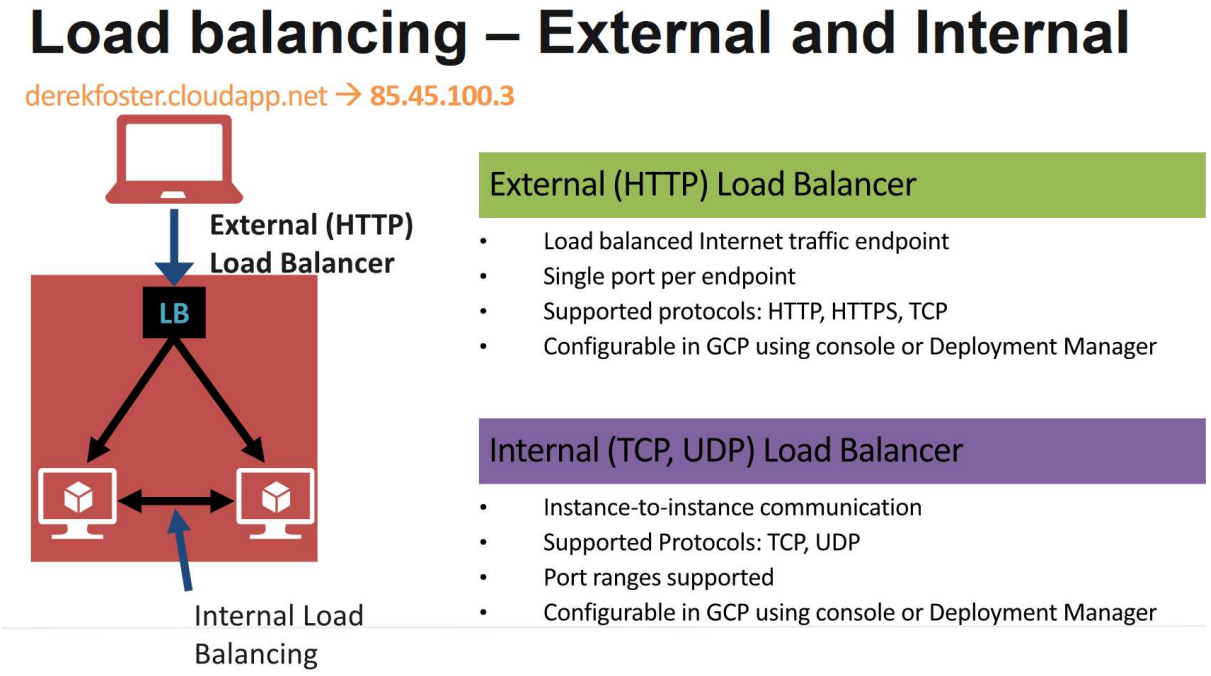
Site-to-Site VPNs

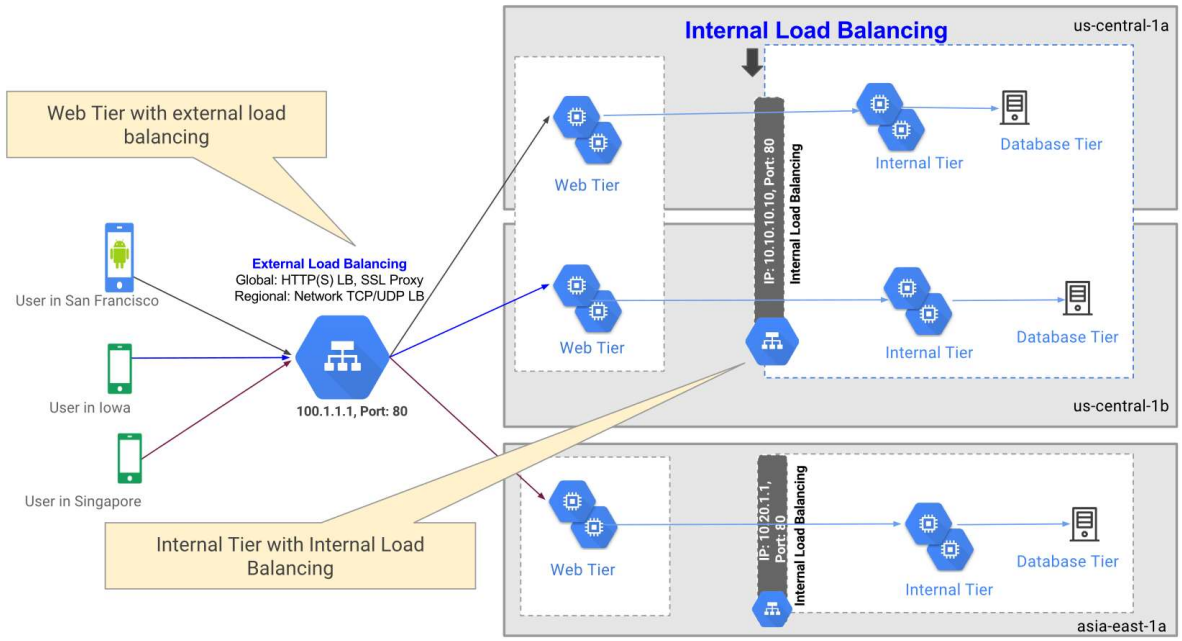


Load Balancing:

- Load balancing is very similar across all cloud vendors
- Distributes the traffic/compute resources between instances
- GCP offers three main types (primarily external and internal):
    - Global (external) load balancing
    - Regional (external) load balancing
    - Internal load balancing
- You must have at least two instances to load balance







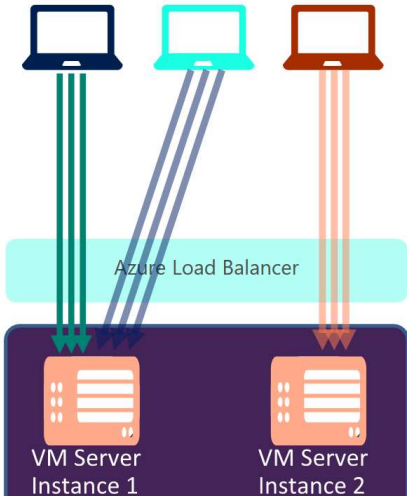
GCP & Azure Load Balancing Algorithms

Default

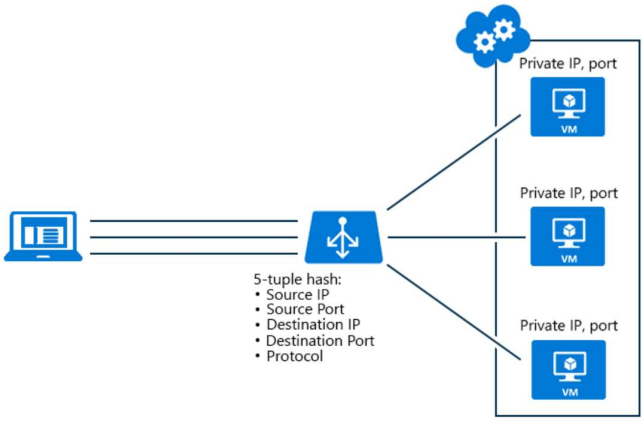
- **5-tuple-hash based**; spreading incoming connections to all active instances

Source-IP-based session affinity

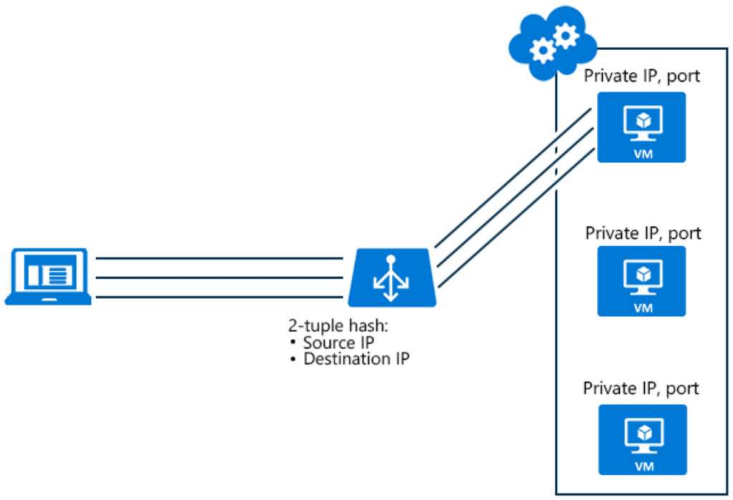
- All connections from the same Internet client IP to the same backend virtual machine server
- 2/3 Tuple
- **Scenarios**
  - Applications that require multiple connections to the same server
  - Example: media streaming to establish control and data channel to same backend server



Default 5-tuple load balancer



Source IP affinity 2-tuple load balancer



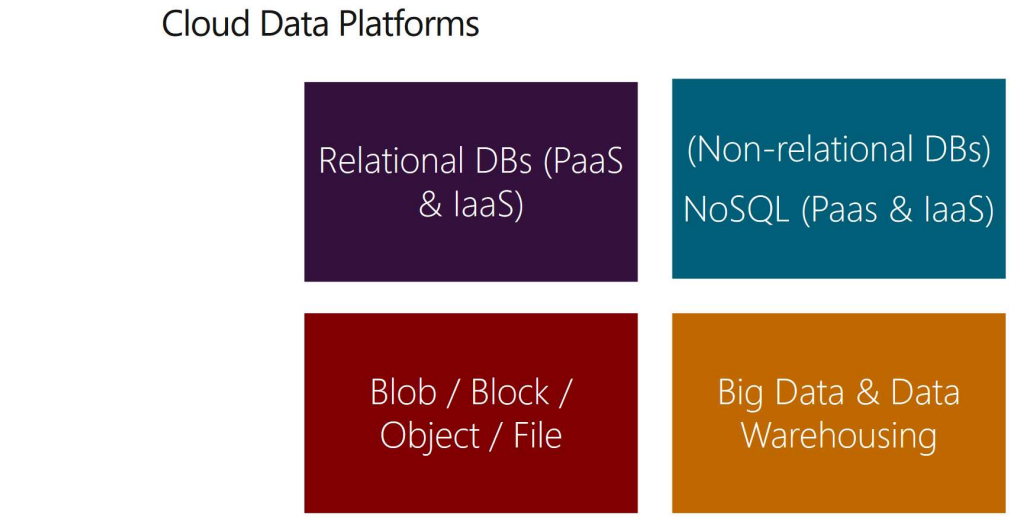
Load Balancing Summary:

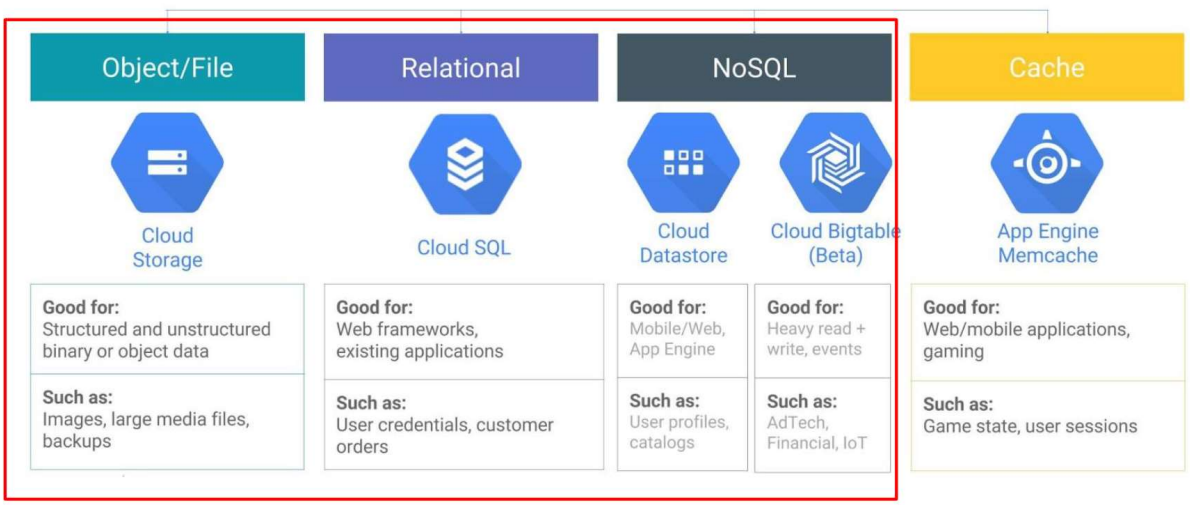
- Load balancing describes how traffic and compute load is distributed between cloud resources
- Two main types: external (HTTP, public internet facing) and internal (private, non-internet facing)
- Requires at least two VM instances to work
- External load balancing is configured when you deploy two or more instances as an instance group on Google Cloud Platform
- Internal load balancing is an optional layer of load balancing, and will need to be configured to be part of an instance group too if required

Cloud Storage

Cloud Data Platforms:

- Data platform refers to the high-level database tools/services that manage data
- Data storage refers to how the data is stored e.g. relational/non-relational, structured/unstructured, blob/block, file storage

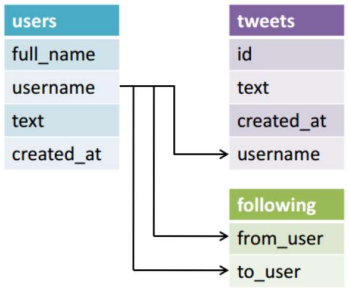


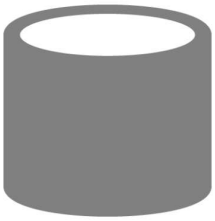


Cloud SQL (PaaS)

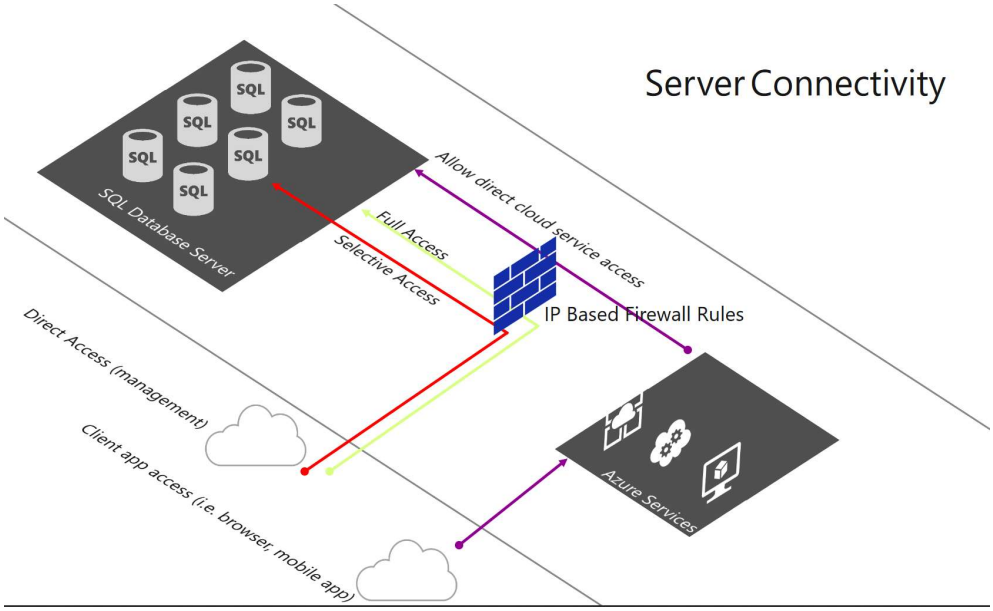
Relational Data Storage

- Relational database as a service
- Your familiar database!
- Primary / foreign keys, tables, GUI designers etc.
- Fully managed
- High availability and scalability
- Automatic backups
- SLA uptime normally 99.99%
- Available through deployment manager as resource type: **sqladmin-v1beta4**

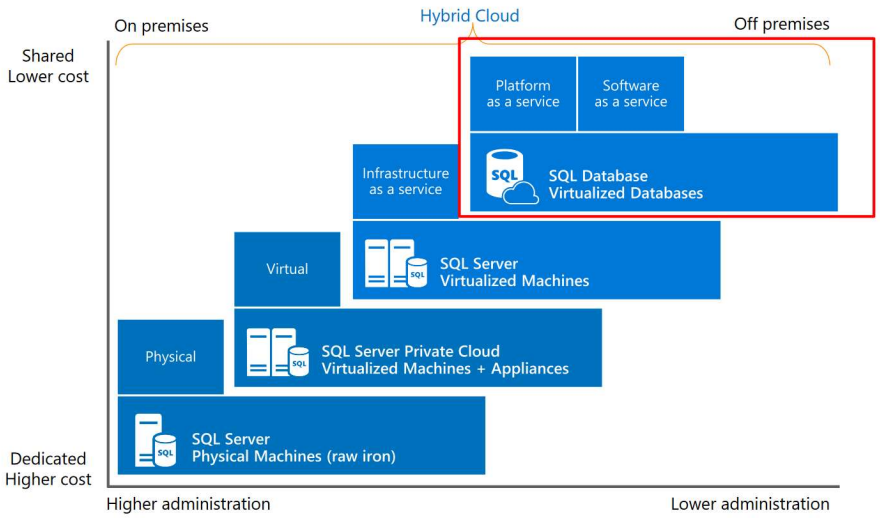




Server Connectivity

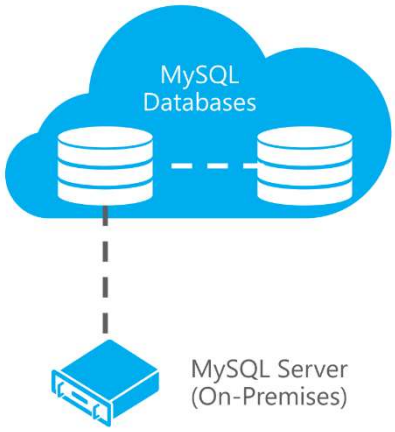


SQL Data Platform – The Business Case



Hybrid Cloud SQL Solution for Business:

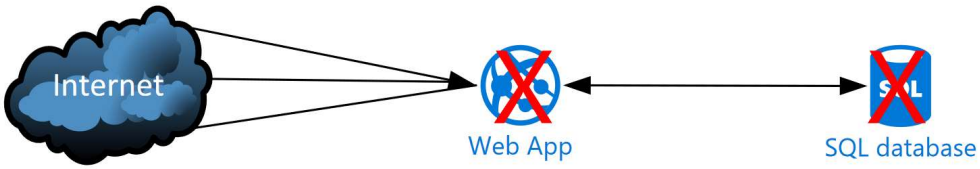
- Synchronize data
- Database synchronization as a service
- On-premise and MySQL Cloud Database
- Provides redundancy and high availability
- Elastically scales with demand
- Synchronizes data between on-premise and cloud
- Extra redundancy



High Availability (HA) Databases:

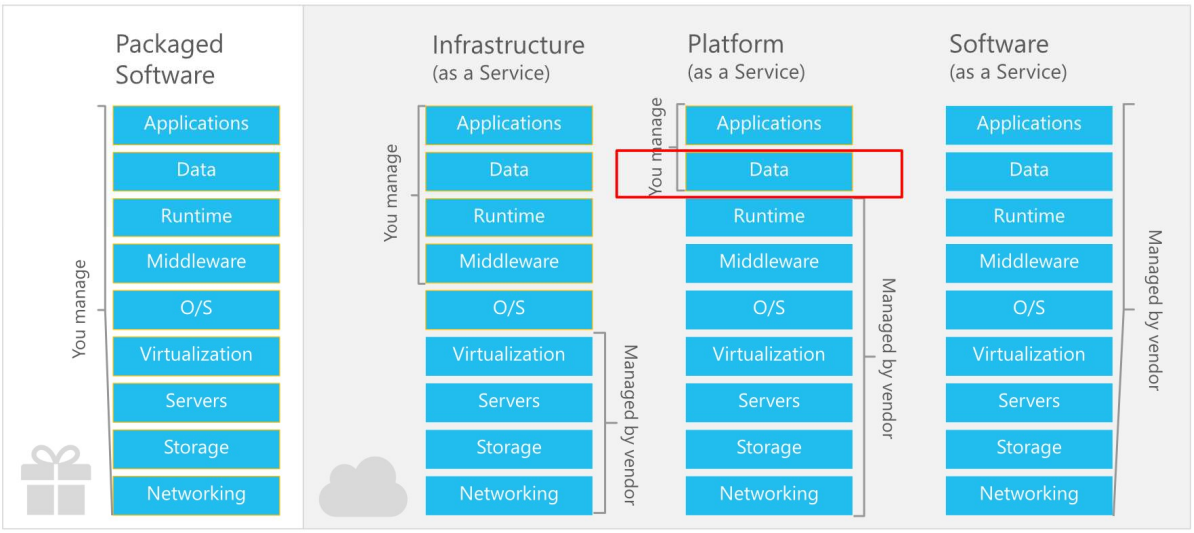
"In cloud terminology, high availability refers to a system or component that is continuously operational for a desirably long length of time. Availability can be measured relatively to '100% operational' or 'never failing'. A widely-held but difficult-to-achieve standard of availability for a system or product is know as 'five 9s' (99.999%) availability" – in the service level agreement (SLA)

High Availability (HA) Databases – non-cloud platform



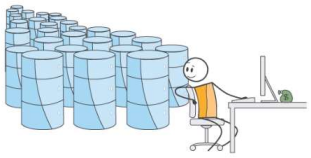
Is this an HA database configuration?

High Availability Data Storage on PaaS



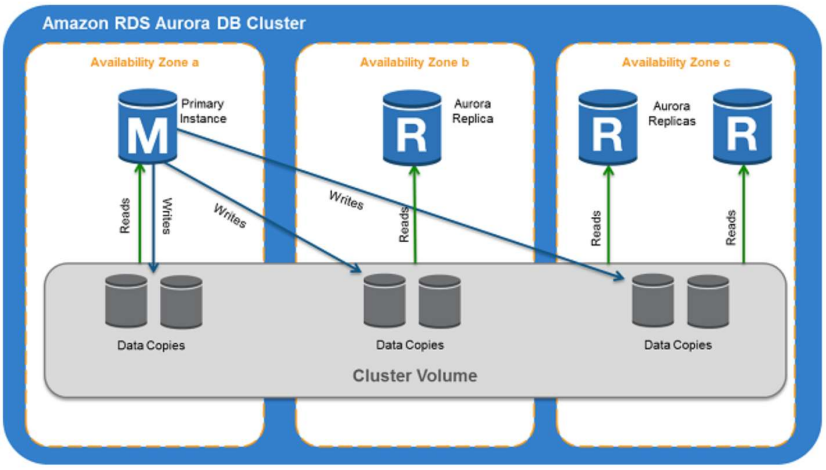
High Availability (HA) Databases – PaaS Replication

- By default SQL (in Azure) or MySQL (in Amazon EC2) PaaS services will have multiple copies created by default
- The will give you HA automatically with all the benefits
- **Only** if you use the PaaS option (not IaaS)
- If you have a single SQL DB on Azure it will have **3 copies** kept in sync, if the primary goes down then one of the copies will take over as primary
- Amazon Aurora (MySQL DB engine) has **6 copies** of each database kept in sync



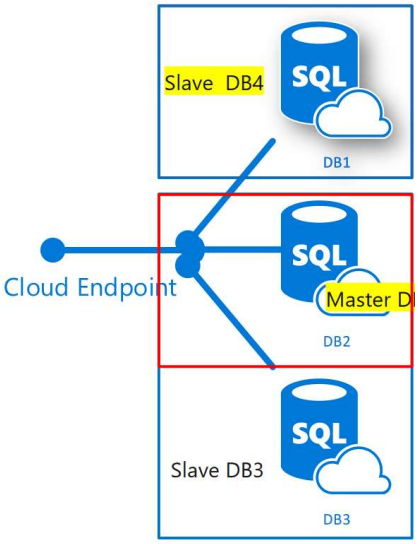
Amazon
Aurora

High Availability (HA) SQL Databases – Amazon Aurora (PaaS)



Wordpress with HA database service

- You want to use MySQL PaaS for your Wordpress site
- Deploy new MySQL PaaS database instance for new Wordpress site via cloud management GUI (few clicks!)
- Developer sees only **one** database instance – the Master DB
- However, there are at least **two Slave copies** of the Master on other cloud servers, possibly in other Datacentres for extra redundancy
- If Master DB goes down, one of the Slave copies becomes the new Master DB, and a new Slave copy is deployed to maintain a standard level of redundancy



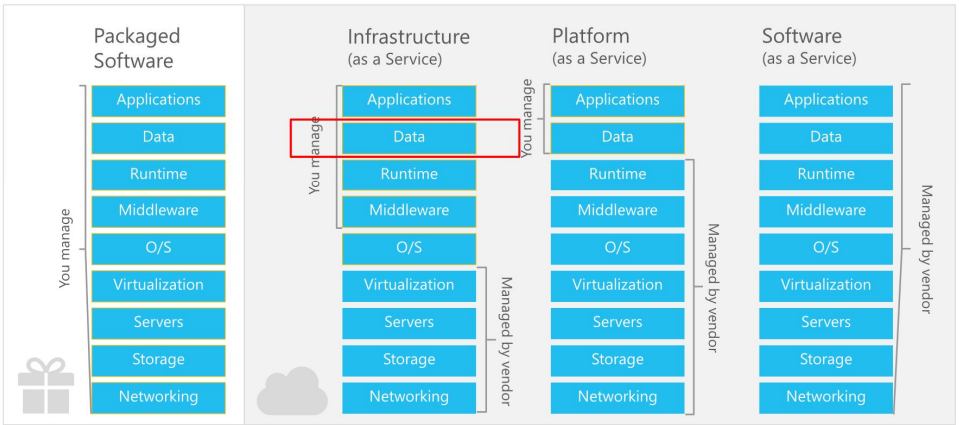
HA SQL Databases – PaaS Geo-Replication

- Sometimes you need iron-clad redundancy
- Normally SQL database copies are ‘locally redundant’, essentially in the same datacentre
- Geo-replication has copies of your database synced to other regions of the world which is helpful in scenarios such as a natural disaster
- Failover hands database connectivity to secondary region

High Availability (HA) SQL Databases – PaaS **Geo-replication**

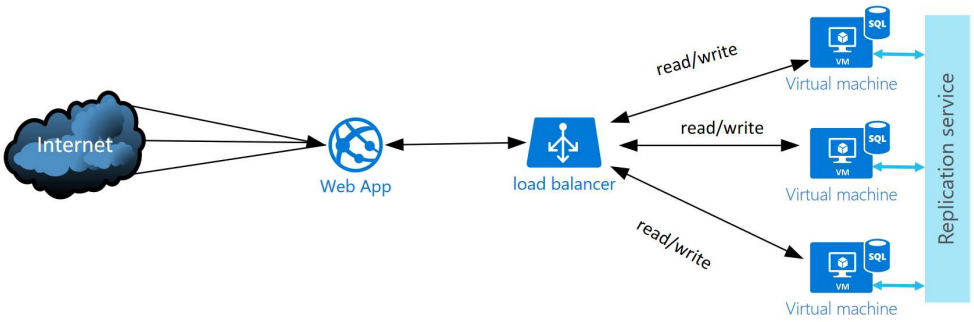
Primary database in Europe with four readable secondaries synced in distinct geographical regions

High Availability Data Storage on IaaS



HA Databases – IaaS Replication:

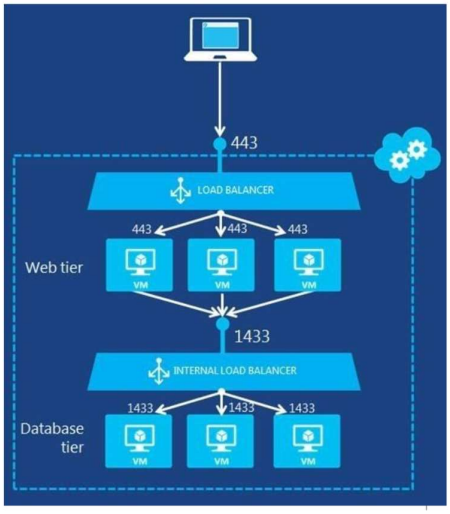
- It is common to require more control and portability with data management – sometimes PaaS doesn't provide the required complexity or control
- Configuring and deploying virtual machines with SQL server running is a common approach
- However, setting up a HA service across your own virtual machines is a difficult task – you have to do everything from choosing the OS, installing the SQL server service, run backups, and configure responsive replication for syncing data read/writes to all database instances

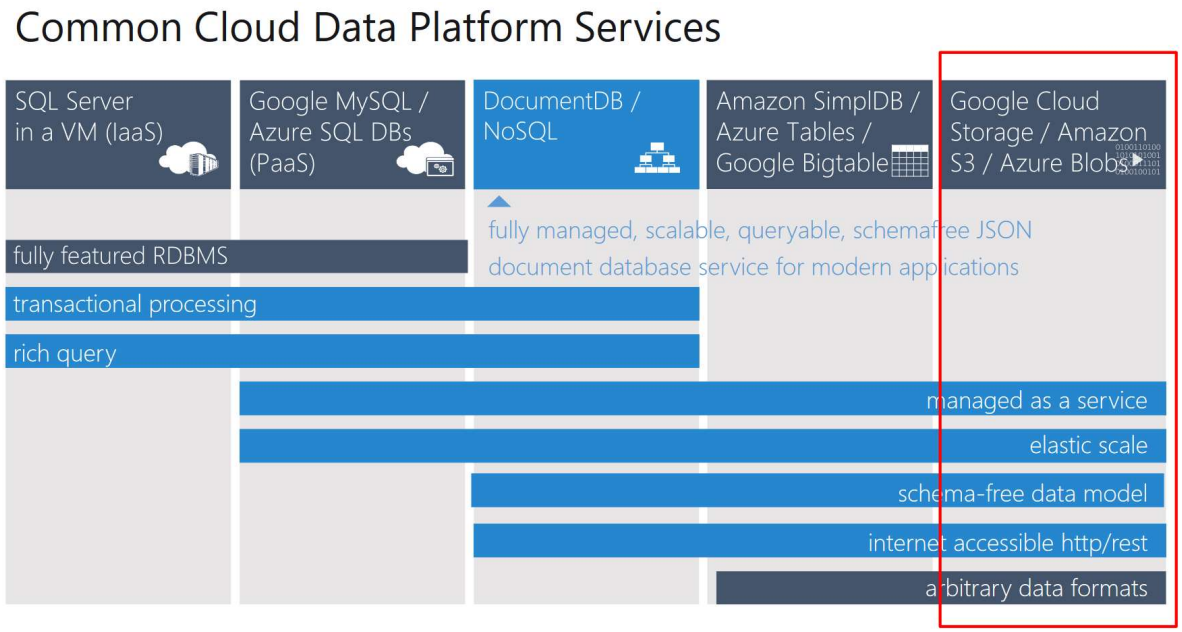


- There is no replication of the **data writes** between the virtual machines running SQL server – writes not synced!
- Needs a replication service running on each server

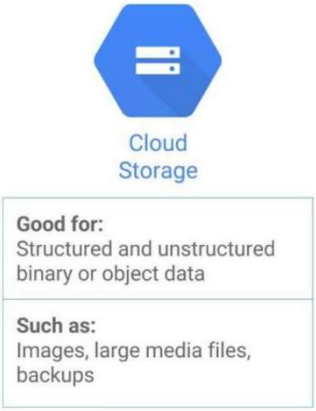
High Availability (HA) Databases – IaaS Replication

- Diagram on right shows a high availability web service with redundancy across both the web and data tiers
- External load balancer will distribute internet traffic across the web tier virtual machines
- Internal load balancer will distribute database I/O requests across data tier virtual machines running SQL server
- Replication service running on all SQL VMs will sync writes



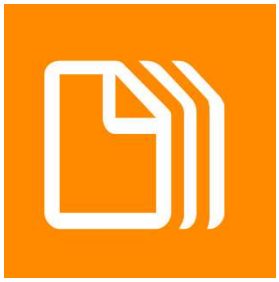


Cloud Storage Fundamentals:



- Simple cloud storage is inexpensive, you can have lots of it for little cost – great advancements in storage tech
- This is readily seen in the fact that many cloud storage services like Dropbox, Google Drive, One Drive, are free
- Across cloud vendors/platforms, the underlying mechanisms for large-scale unstructured data storage is very similar
- Can store anything from text files, executables, to virtual images – very diverse

Unstructured Data Storage:



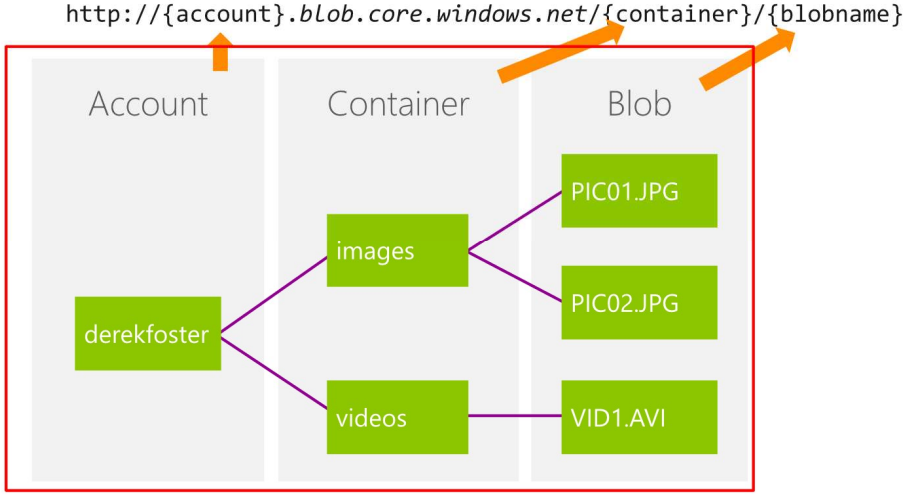
- Cloud Storage in GCP (Binary Large Object in Azure, Simple Storage S3 in Amazon)
- Hundreds of gigabytes per blob in size
- 100TB+ per storage account possible
- REST API
- Geo-replication for disaster recovery
- Dropbox lived on Amazon's S3 for 8 years

Object/Blob/Simple Storage (Bucket):

- Google Cloud Platform and Amazon's simple storage services stores unstructured data in a bucket
- Azure's Blob storage service stores unstructured data in a container as objects/blobs
- All vendor's storage can store any type of text or binary data, such as a document, media file, or application installer
- This type of storage is fundamental to the cloud
- You can have a number of blob/simple storage accounts and it makes sense to do so for different services

Azure Blob Storage Concepts

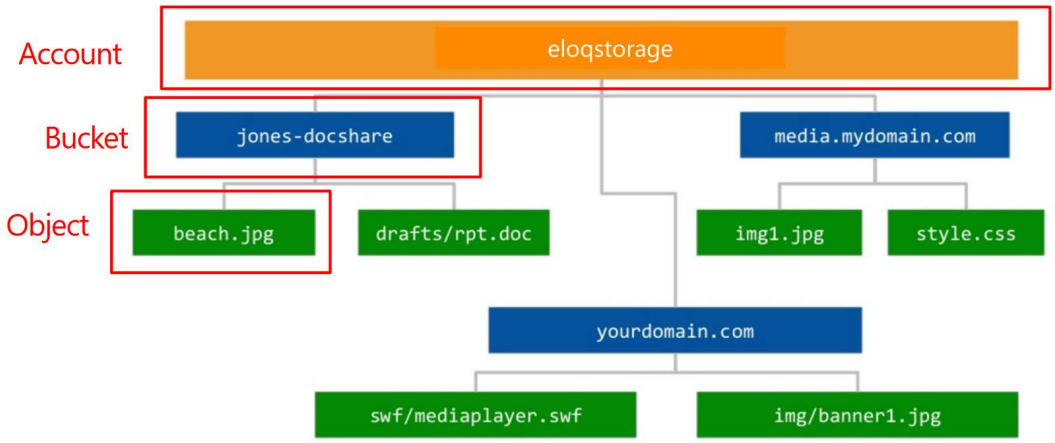
http://{account}.*blob.core.windows.net*/{container}/{blobname}



http://derekfoster.*blob.core.windows.net*/images.PIC01.JPG

Amazon Simple Storage Concepts

http://{account}.*amazonaws.com*/{bucket}/{object}



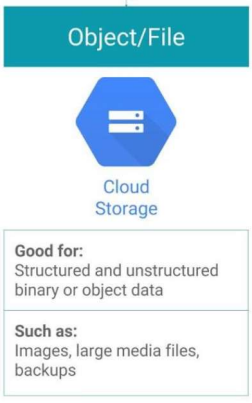
http://eloqstorage.amazonaws.com/jones-docshare/beach.jpg

Google Cloud Storage

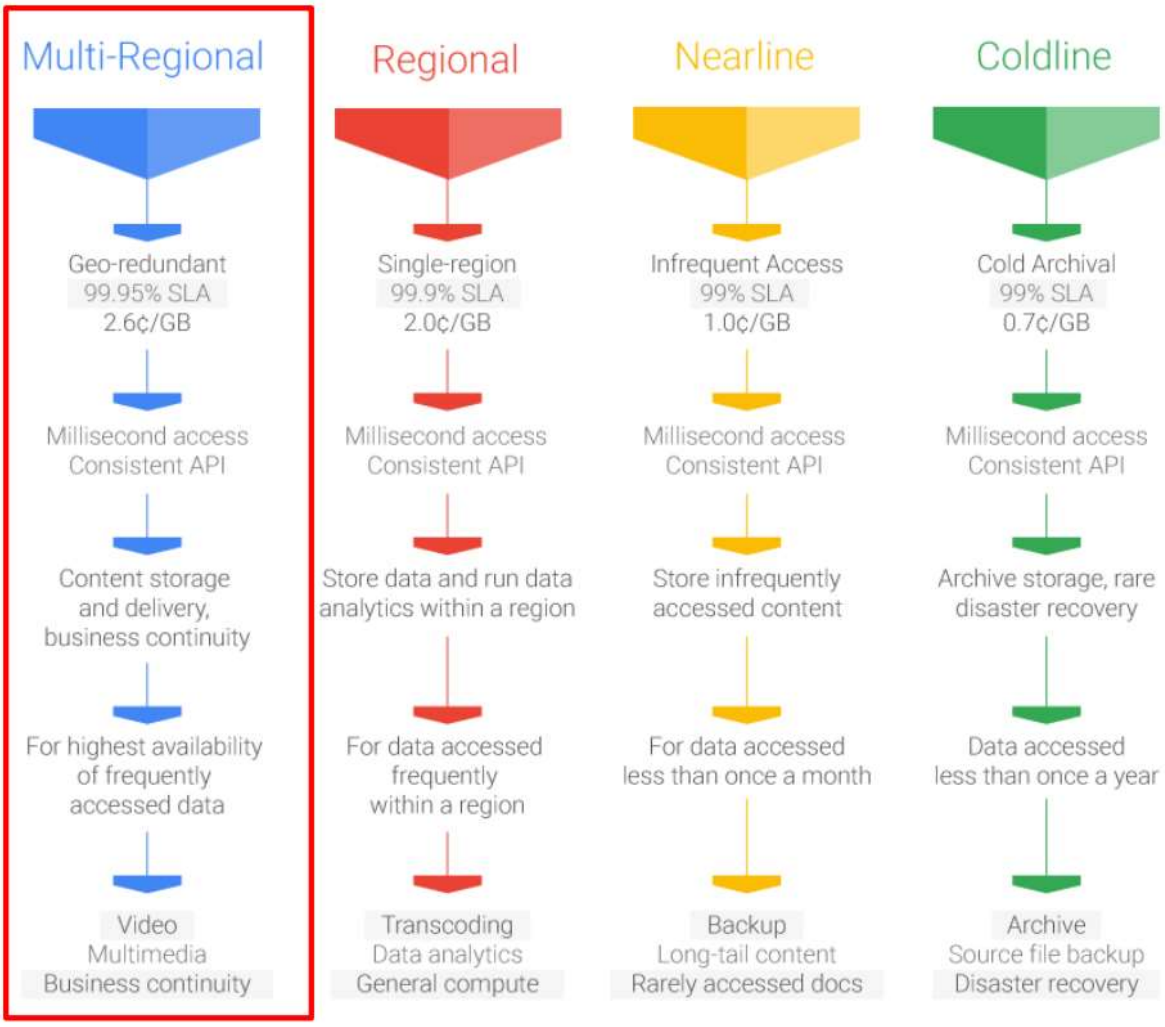
https://www.googleapis.com/storage/v1/b/[BUCKET_NAME]/o/[OBJECT_NAME]

- As with Azure and Amazon, Google's cloud storage is a RESTful online storage service
- Can be used for blobs, objects, files etc.
- Google Buckets can be deployed via deployment manager using resource type:

  storage.v1.bucket



Google Cloud Storage Classes

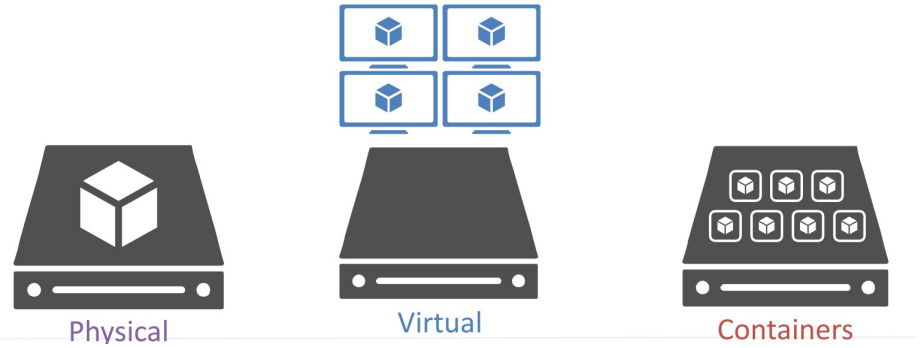


Container Computing

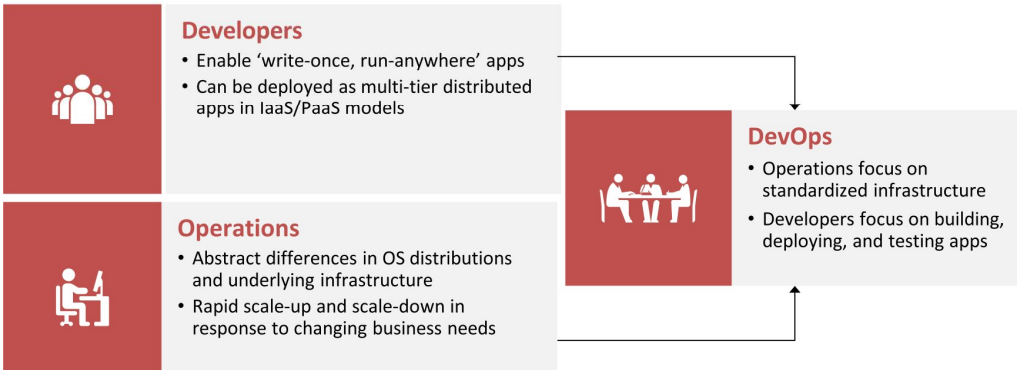
Containers:

- Container computing is commonly referred to as container-as-a-service (CaaS)
- Containerised applications run in the cloud and are used by consumers
- Many of Google's services such as Gmail and YouTube are examples of containerised web apps
- Every time you use Gmail or YouTube, an instance of a container for the app is run – Google has the capacity to run billions of such app containers
- Companies can develop their apps in-house then containerise them for deployment on cloud and mass consumption
- Containerised apps are easy to move between cloud vendors

A new approach to build, ship, deploy, and instantiate applications



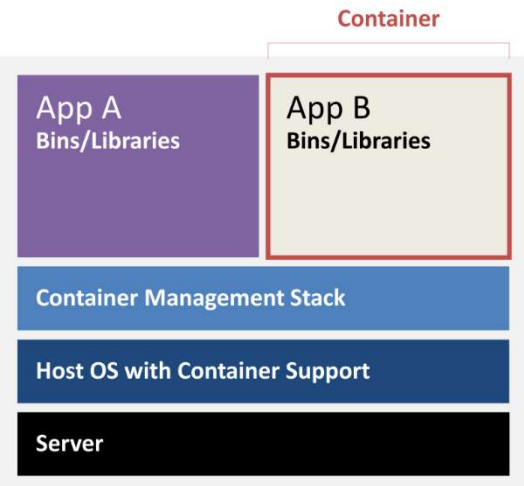
Why containers?



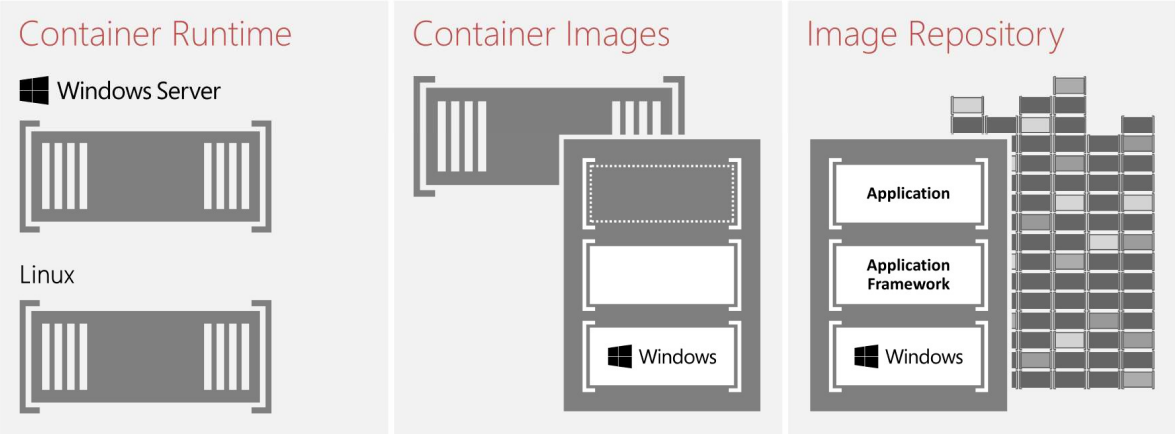
Containers

- Dependencies – every application has its own dependencies which includes both software and hardware

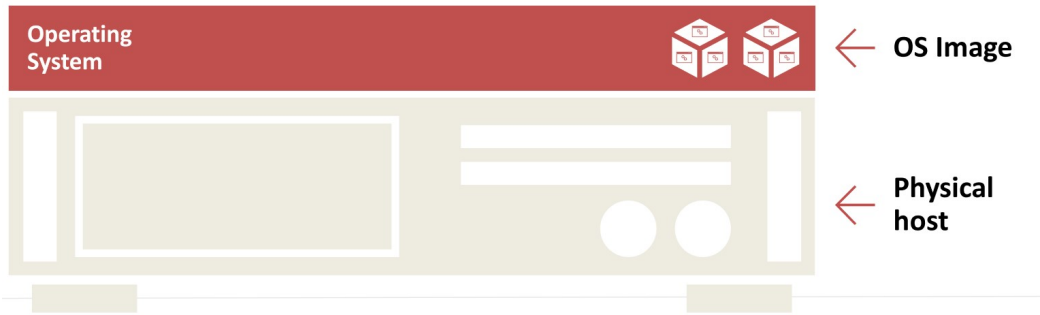
- Virtualisation – container engine is a light-weight virtualisation mechanism which isolates these dependencies per app
- Shared host OS – container runs as an isolated process in user space on the host OS, sharing the kernel with other containers
- Flexible – differences in underlying OS and infrastructure are abstracted away, a 'deploy anywhere' approach
- Fast – containers can be created almost instantly, enabling rapid scale-up and scale-down in response to changes in demand



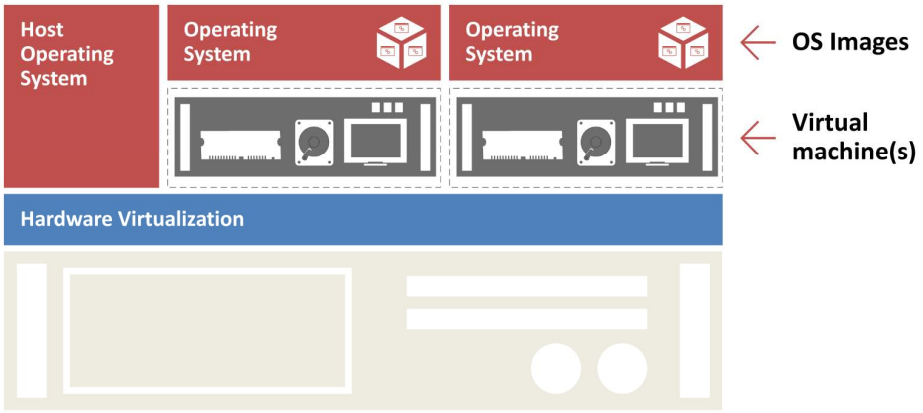
Container ecosystem



Pre-virtualisation

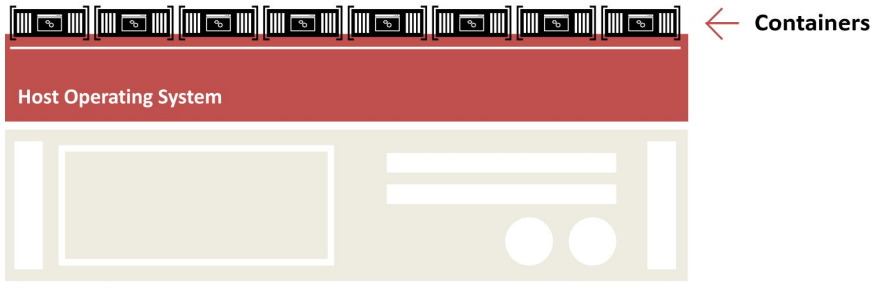


Server Virtualisation



Container runtime on a physical server

- Containers “share” the server host operating system but have isolated processes.



Container runtime within a VM

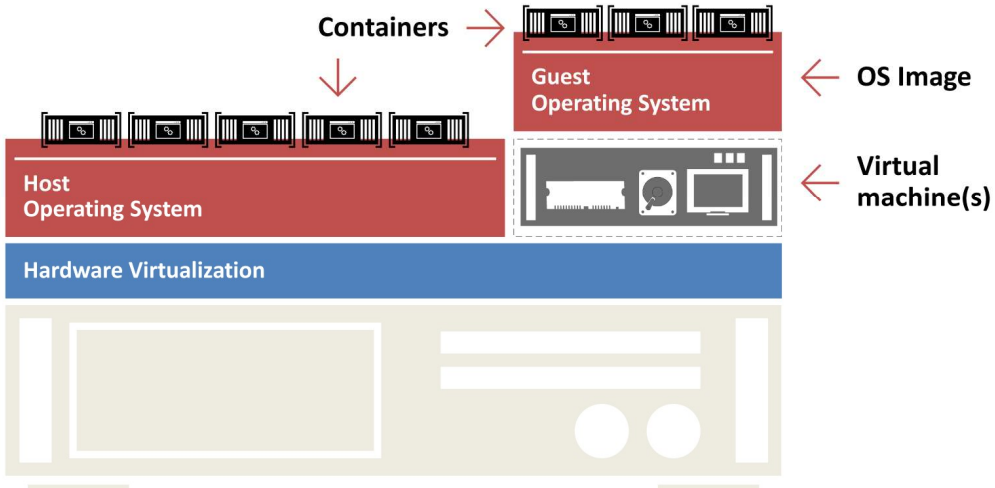
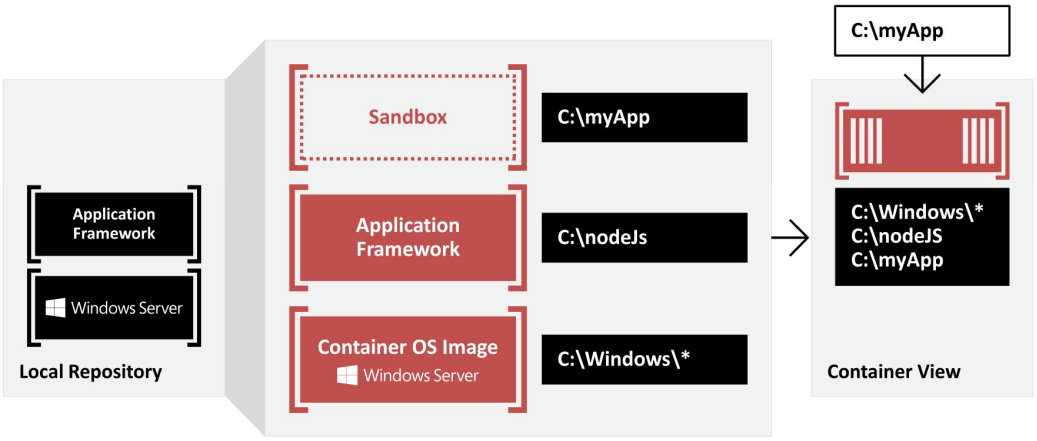
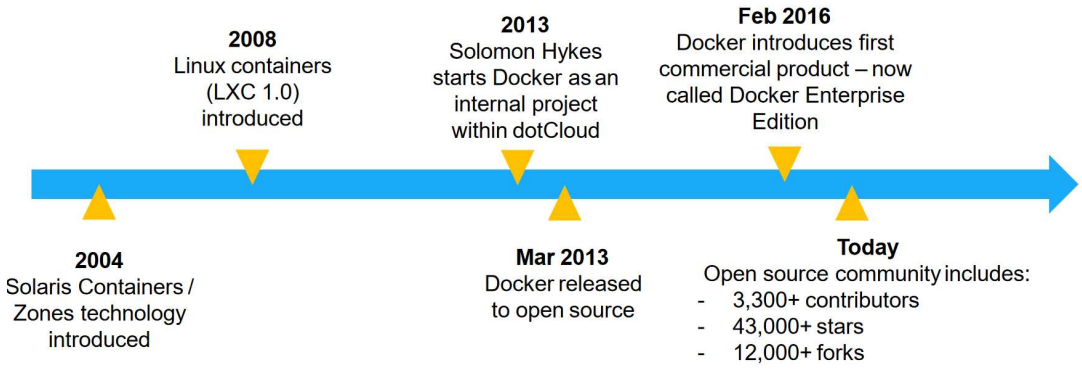


Image Creation

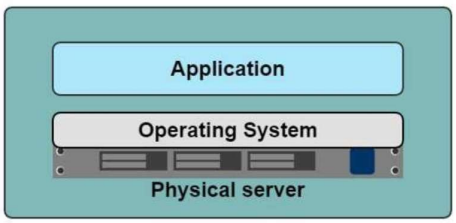


History of Docker



Significant adoption

One application on one physical server

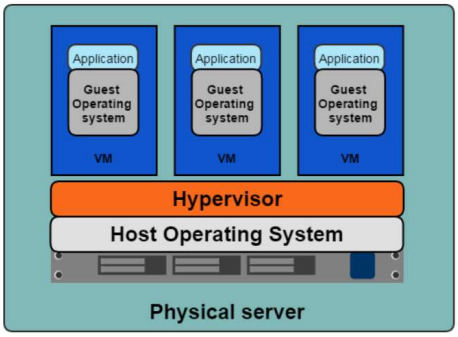


Historical Limitations of application deployment:

- Slow deployment
- Huge costs
- Wasted resources
- Difficult to scale
- Difficult to migrate
- Vendor lock in

Hypervisor-based Virtualization

- Can contain multiple applications
- Each application runs in a virtual machine (VM)

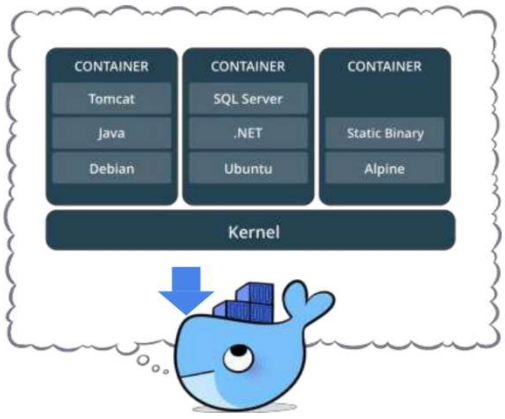


Limitations of Virtual Machines:

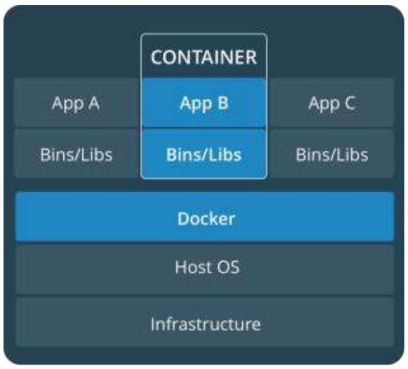
- Each VM still requires; CPU allocation, storage, RAM, an entire guest operating system
- The more VMs you run, the more resources you need
- Guest OS means wasted resources
- Application portability not always guaranteed (but commonly is)

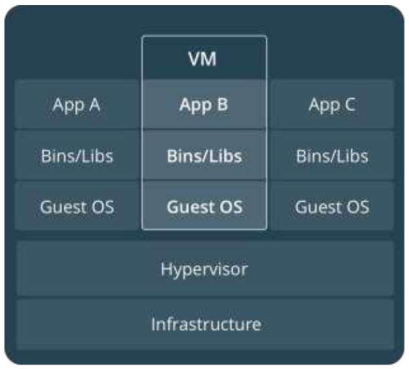
What is a Docker Container?

- Standardised packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works with all major Linux and Windows OS
- Can run Windows and Linux containers on a Windows Server kernel



Comparing Containers and VMs

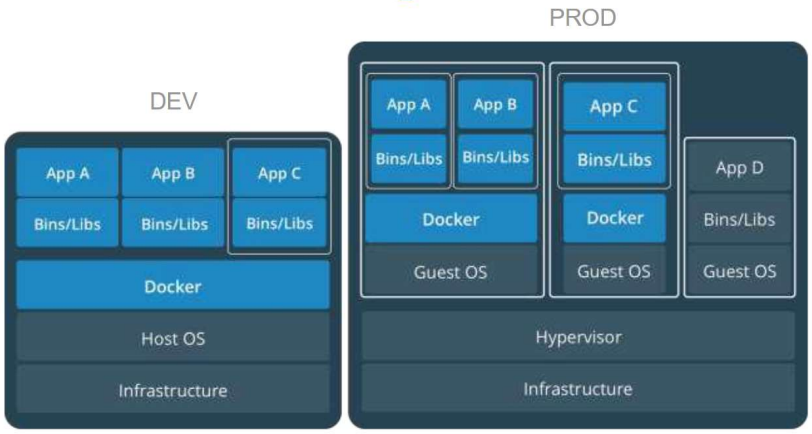




Containers are an app level construct

VMs are an infrastructure level construct to turn one machine into many servers

Containers and VMs together



Containers and VMs together provide a tremendous amount of flexibility for IT to optimally deploy and manage apps.

Lesson Key Benefits of Docker Containers

Speed

- No OS to boot = applications online in seconds

Portability

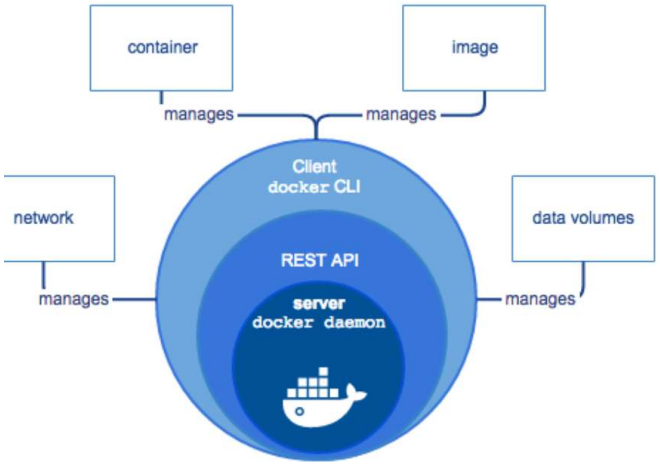
- Less dependencies between process layers = ability to move between infrastructure

Efficiency

- Less OS overhead
- Improved VM density

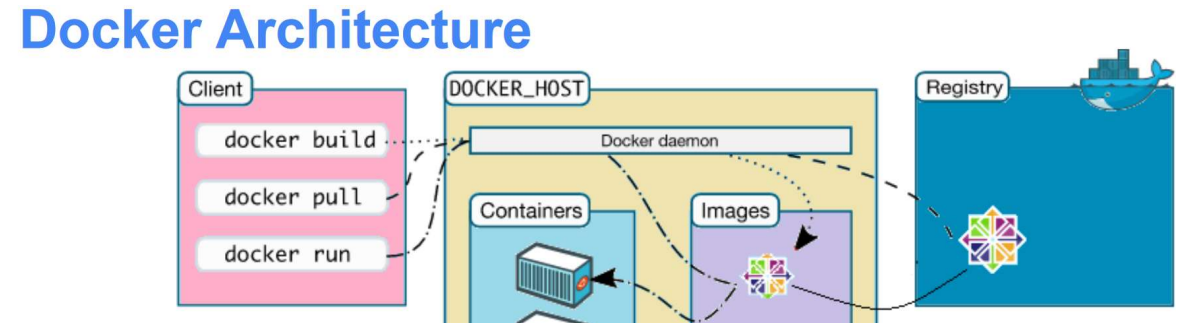
Docker Architecture:

- Docker uses a client-server architecture
- The docker client talks to the Docker daemon, which does the majority of the building, running, and distributing docker containers
- The docker client and daemon can run on the same system, or you can connect a docker client to a remote docker daemon (e.g. running in the cloud)
- The docker client and daemon communicate using a REST API, over UNIX sockets or a network interface



Docker Engine – a closer look:

- A server which is a type of long-running program called a daemon process
- A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do
- A client command line interface (CLI)
- The docker daemon creates and manages docker objects, such as images, containers, networks, and volumes
- The user will generally manage docker objects through the client CLI



The **Client** and **DOCKER_HOST** may be installed on a local machine, or:

- Client installed locally with DOCKER_HOST in cloud
- Both Client and DOCKER_HOST in cloud
- **Registry** service normally in cloud

Objects – Docker Images:

- An image is a read-only template with instructions for creating a docker container. Often, an image is based on another image, with some additional customisation. For example, you may build an image which is based on the ubuntu image, but installs the apache web server and your application, as well as the configuration details needed to make the application run
- You might create your own images or you might only use those created by others and published in a registry. To build your own image, you create a dockerfile with a simple syntax for defining the steps needed to create the image and run it

Objects – docker containers:

- A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the docker API or CLI. You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state
- By default, a container is relatively well isolated from other containers and its host machine
- A container is defined by its image as well as any configuration options you provide to it when you create or start it. When a container is removed, any changes to its state that are not stored in persistent storage disappear

Docker Terminology Summary

Docker Image
The basis of a Docker container – represents a full application at rest

Docker Container
The image when it is ‘running.’ The standard 'unit' for an app service

Docker Engine
Creates, ships and runs Docker containers deployable on a physical or virtual, host locally, in a datacentre or cloud service provider

Registry Service
Stores, distributes and manages Docker images

Cloud Computing Ethics & SLAs

Ethics:

- Firstly, ethics refers to well-informed standards of right and wrong that prescribe what humans ought to do, usually in terms of rights, obligations, benefits to society, fairness, or specific virtues
- Secondly, ethics refers to the study and development of one's ethical standards. As mentioned above, feelings, laws, cultural and social norms can deviate from what is ethical. So, it is necessary to constantly examine one's standards to ensure that they are reasonable and well-founded

"Technology is not Neutral"

Examples of Computing Ethical Scenarios:

- Peer-to-peer file sharing
- Using dodgy copies of windows/games
- Hijacking neighbours' unsecure internet
- Reusing images/IP from websites
- Letting children play 18 rated games
- Letting children under 13 have a Facebook account
- Developer not encrypting sensitive data
- Machines making humans solve captchas

Computers are…

- Uniquely complex and connected
- Uniquely fast and pervasive
- Uniquely malleable and reprogrammable
- Uniquely cost effective and available to all
- Uniquely adept at producing perfect digital copies

Computer Ethics:

- Some of the common issues arising from these characteristics include privacy, ownership and copyright, and liability
- Some of these are covered by professional ethics, others are covered by personal ethics
- Will it appear insensitive or reckless, or be seen as taking more risk than we should have?
- Does the action violate a company's stated values (written policies)?
- Will you feel bad if you take this action?
- Ask someone you trust for guidance (a friend, the corporate ethics officer)
- Keep asking until you have an answer
- Don't just do what your boss asks you if you're unsure, its your responsibility if things go wrong

Cloud and Data:

- Many of the services we use daily are cloud powered
- Your data is stored and duplicated across datacentres around the world
- Different countries have different laws applied to the data
- Questions of cloud data ownership, security, and privacy are frequent headline grabbers

Cloud Storage Providers:

- Cloud storage is now 'free': Dropbox, One Drive, Google Drive, iCloud
- Is it really free? Do you really own your data?
- Microsoft One Drive Terms of Service…
    - "don't publicly display or use the services to share any inappropriate content or other material (involving, for example, nudity, bestiality, pornography, offensive language, graphic violence, or criminal activity)"
    - By breaching the terms of service your account could be blocked, including linked accounts losing access to files, email, Xbox services, cloud services, Office 365
- Dropbox Terms of Service…
    - "you retain full ownership of your stuff"
    - Data is stored on Dropbox servers
    - Claim they won't share your content
    - Not responsible for loss of corruption of data, nor for any costs of backing up or restoring it
    - Can terminate service at any point without notice, but will 'try' to let you know in advance

Datacentres:

- Over 90% of Facebook datacentres are in the US, despite over 70% of its user-base being outside the US
- All Twitter datacentres are in US
- Majority of Google datacentres in US
- Safe to make the assumption that most of your data used by these services will not be under US jurisdiction, not UK

Cloud Computing and Ethics:

- Connectivity alternatives
- Datacentres: owned or rented, security, physical location, governing laws
- Vendor's ability and policies to assure confidentiality and security
- Unclear policies about data ownership
- Policies for data breach notice
- Assurance of data destruction upon termination
- As a developer, it is important to be aware of the potential ethical implications of using cloud providers

- Transparency is good, clear terms of service is key

Cloud Providers:

Read Terms of Service (ToS) or Service Level Agreement (SLA) for…

- Data ownership
- Confidentiality (and information sharing with third parties)
- Data destruction policies
- Business interruption and continuity (uptime)
- Data access
- Security policy

Important in context of privacy / important for business continuity in the event of loss of service

Data Ownership:

- The thorny issue of data ownership has caused businesses and developers to wonder if its safe to turn their data over to cloud providers and trust it off-site on someone else's servers
- Some cloud providers say that data ownership isn't even a question, the customer owns it, no 'ifs' or 'buts'. But it is up to the developer and cloud provider to spell out exactly what it has access to and what happens in the event that data is unavailable or inaccessible

Confidentiality:

- Will your organisation's details or data be shared with any third parties (e.g. law enforcement)?
- Are there geographical variances in data privacy laws where your data is stored – different jurisdictions and law?

- Are there any providers' tools deployed that automatically scan your data in the cloud?
- Google mines all your data
- Microsoft One Drive deploys image detection
- Could these compromise your organisations or customers confidentiality?

Data Destruction:

- You may need to comply with local government laws on data storage and destruction, particularly for sensitive data
- How long can you keep it for in the cloud? Is it easy to move the data to another provider?
- If you close your account with the cloud provider, is your data really destroyed immediately?
- Are there any copies kept and for how long?
- Can you request an inventory for your assets in the cloud?

Security:

- Will the provider notify you of any security breach immediately?
- If there is a breach…
    - Has your organisations' IP been infringed?
    - Has your customers' data been compromised?
    - Do you have a strategy?
    - Compensation?

Summary:

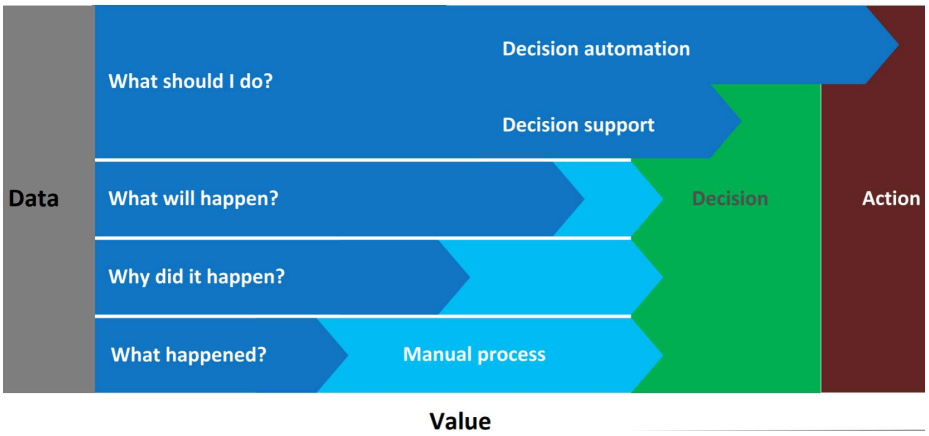
- A lot hinges on the cloud provider to meet the ethical, moral and business expectations of an organisation
- Implications for multiple stakeholders: end users (customers/business), developers, cloud providers
- Read SLAs and ToS

Big Data

Data and Data Science:

- Data science is the exploration and quantitative analysis of all available structured and unstructured data to develop understanding, extract knowledge, and formulate actionable results
- The cloud provides powerful on-demand tools for data sense-making
- Huge industry demand for data scientists, almost all will have expertise in cloud-based tooling

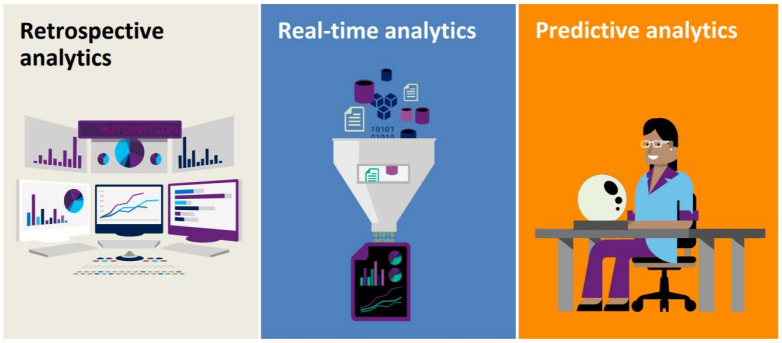
Data → Decisions → Actions



Data Scientists…

- Find/generate data sources
- Acquire data
- Clean and transform data
- Understand relationships in data
- Deliver value from data
- Visualise the result

What Type of Data Analytics?

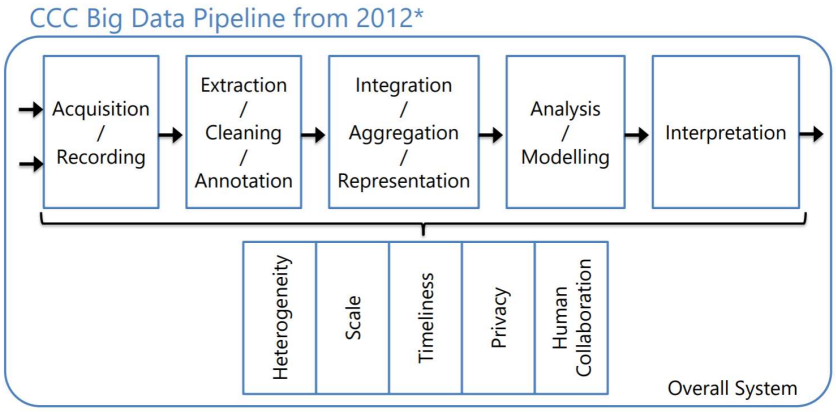


Predictive vs Prescriptive Analysis:

- Predictive analytics calibrated on past data tell us what to expect
- Prescriptive analysis tells us what action to take
- Both are useful for business: finance (markets), social media (advertisement), engineering (failure actions)
- All generated from understanding large volumes of data

Historical Notes – 2012 Big Data Pipeline

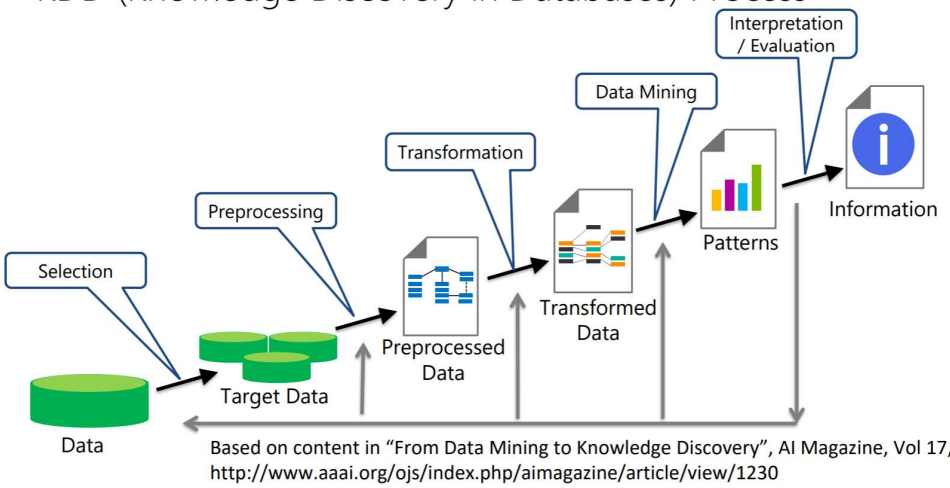
- Term "Big Data" coined by astronomers Cox and Ellsworth in 1997



"NASA researchers Michael Cox and David Ellsworth use the term 'big data' for the first time to describe a familiar challenge in the 1990s: supercomputers generating massive amounts of information — in Cox and Ellsworth's case, simulations of airflow around aircraft — that cannot be processed and visualized. 'Data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk,' they write. We call this the problem of big data."

Historical Notes - 1996

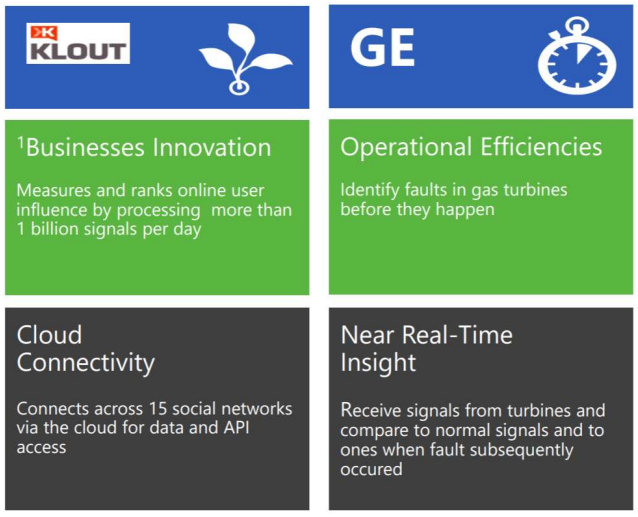
- KDD (Knowledge Discovery in Databases) Process



- The stages of data analysis are basically the same no matter who invents or reinvents the (knowledge discovery / data mining / big data / data science) process.
- Big data analysis is an iterative process

New Business Questions

Bigdata Creates new Opportunities

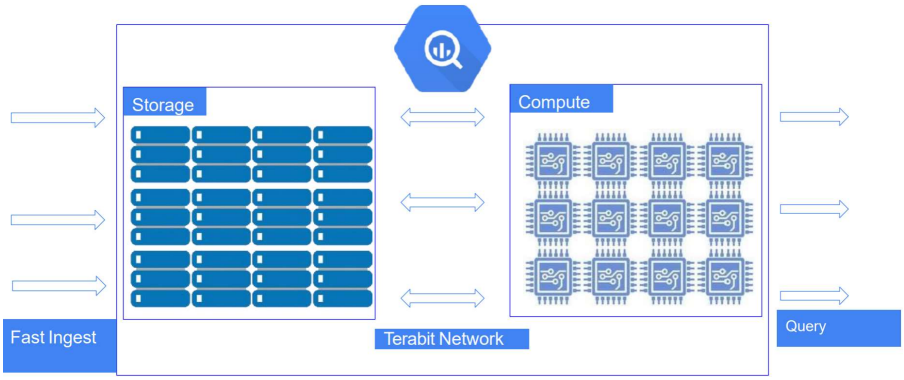


- There are around 30 billion pieces of content shared on Facebook every day. 300 million of those are photos
- Wal-Mart handles more than 1 million customer transactions an hour
- More than 5 billion people are texting, calling, tweeting, and browsing using smart phones

The 3 Vs of Big Data:

- It is a new set of approaches for analysing data sets that were not previously accessible because they posed challenges across one of more of the '3 Vs' of Big Data
- Volume – too big: terabytes and more of credit card transactions, web usage data, system logs
- Variety – too complex: truly unstructured data such as social media, customer reviews, call centre records
- Velocity – too fast: sensor data, live web traffic, mobile phone usage, GPS data

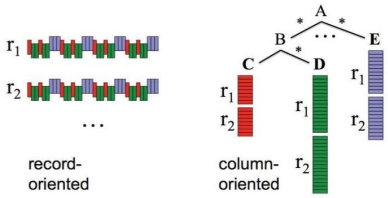
Google – BigQuery

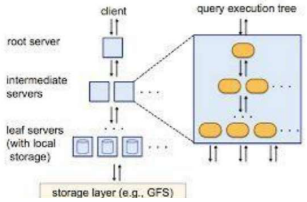


BigQuery – Built on Dremel Architecture

Data Model/Storage:

- **Columnar** Storage
- No Index!
  -> Single Full Table Scan (from disk)



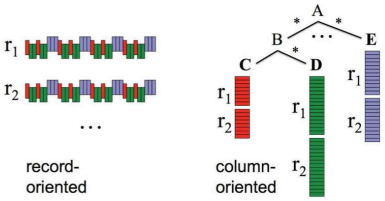


Query Execution:

-**Tree** architecture
- Using tens of thousands machines over fast Google network (+1Petabit/s)

Columnar Storage

- Traffic minimization:
  - Only read selected columns
  - Not the entire record/table



- Faster execution:
  - 85 billion records from 5000 seconds to 30 seconds

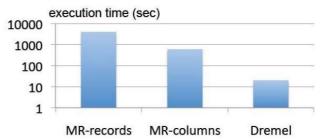
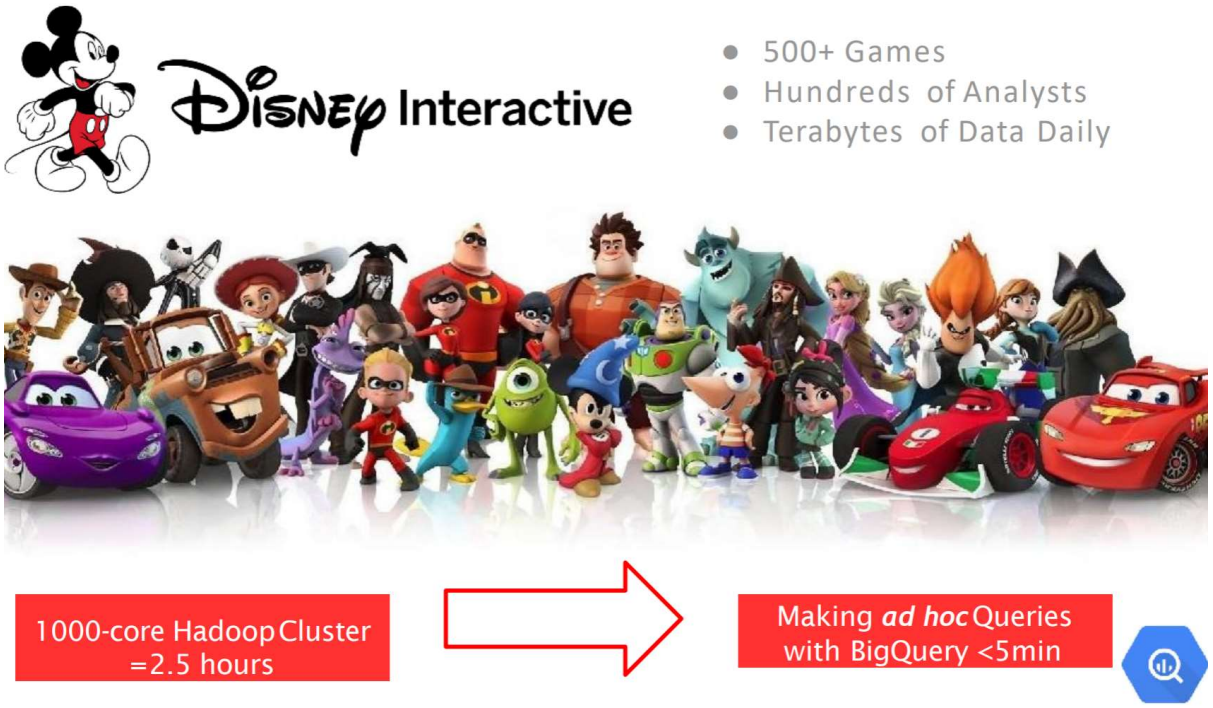


Figure 10: MR and Dremel execution on columnar vs. record-oriented storage (3000 nodes, 85 billion records)

BigQuery Demo showed an inefficient regular expression running on over 100 billion rows from a table 7TB in size, in less than 40 seconds.

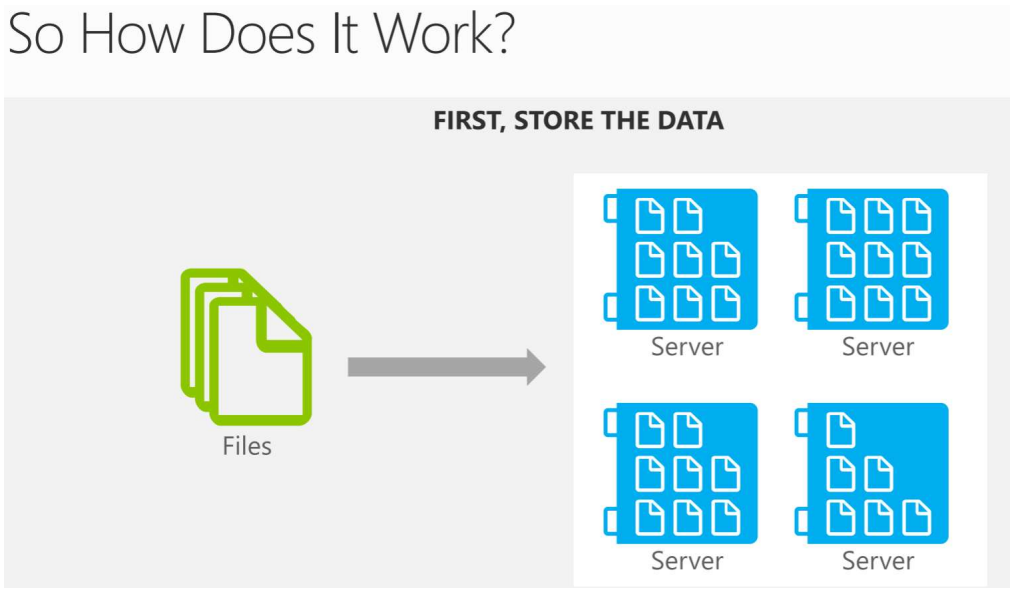
Hardware requirements…

- 330 100MB/sec dedicated hard-drives to read 1TB of data
- 330 gigabit network to shuffle the 1TB of data
- 3,300 CPU cores to uncompress 1TB of data and execute 100 billion regular expressions with 3 wildcards each



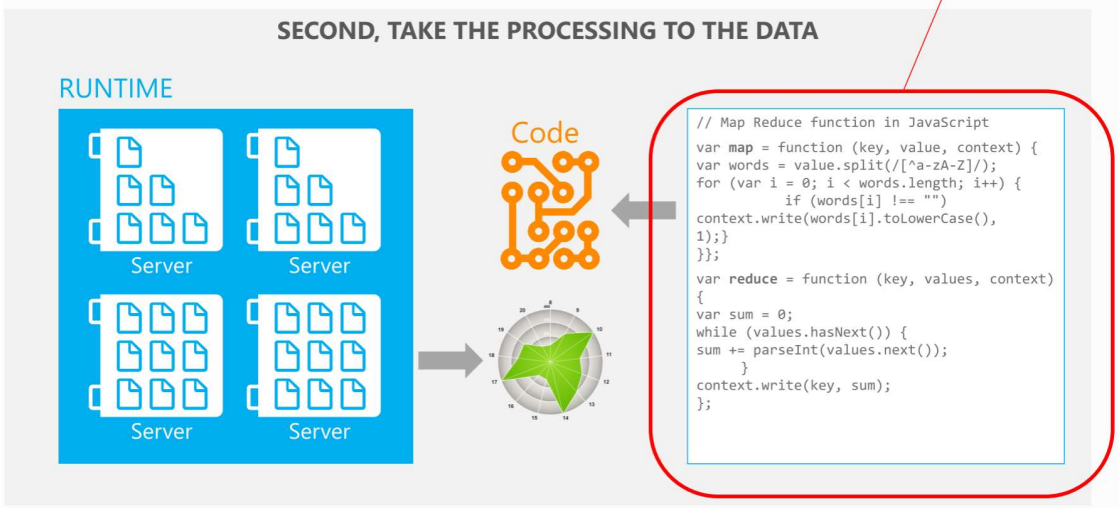
Hadoop:

- Solution: move compute to data
- Google App Engine: appengine-mapreduce API



So How Does It Work?

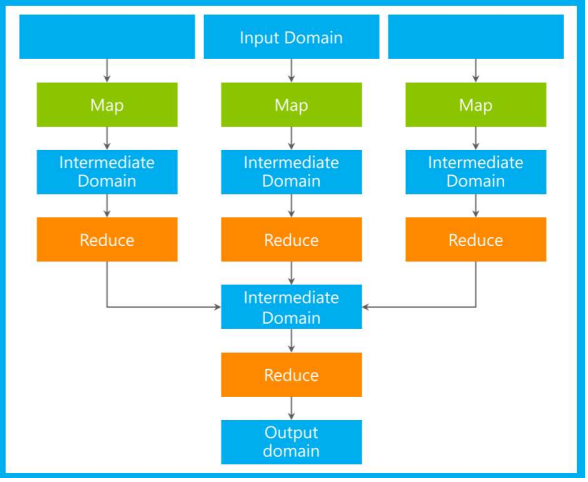
Could be R or Python!



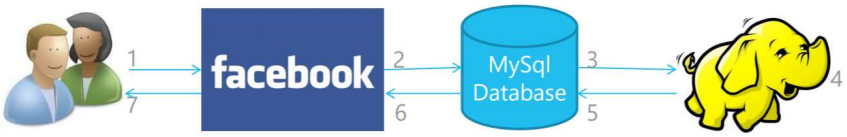
MapReduce – Workflow

A MapReduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner

The framework sorts the outputs of the maps, which are then input to the *reduce* tasks



Example of Bigdata Analytics using Hadoop



1. Users interact with Facebook using data in textual, image, video formats
2. Facebook transfers the core data to MySQL databases
3. My SQL data is replicated to Hadoop clusters
4. Data is processed using Hadoop MapReduce functions
5. The results are transferred back to MySQL
6. Facebook uses the data to create recommendations for you based on your interests

Other users:

Why should enterprises move to Big Data Analytics?

- Enterprises will be able to harness relevant data and use it to make the best decisions
- Determine optimum prices
- Calculate risks quickly and understand future possibilities to mitigate risk
- Enable new products
- Identifying patterns helps identify trends in business