QA Consulting

Introduction to Cloud Computing.

TRAINING MATERIALS - COURSE HANDOUT

Contacts

devdatta.gonsai@qa.com

team.qac.all.trainers@qa.com

www.consulting.qa.com

Contents

What is the Cloud? – 3

Characteristics of Cloud Computing – 3

On-demand Self-service – 4

Broad network access – 5

Resource pooling – 5

Rapid Elasticity – 6

Measured Service – 6

Benefits of the Cloud – 6

Cloud Model of IT Provisioning – 7

Cloud Benefits – 7

High Availability – 8

Agility – 8

Cloud-Enabling Technologies – 9

Fast, wide area networks – 9

Powerful, inexpensive server computers – 9

High-performance virtualisation – 9

Benefits of virtualistion – 10

Virtualisation vs Multi-tenanting – 10

Containers – 11

Services – 12

Service-Oriented Architecture – 12

SOA and the Cloud – 13

Microservices – 13

Microservices Example: Amazon Homepage – 14

SOCCI: Service-Oriented Cloud Computing Infrastructure – 15

IaaS, PaaS and SaaS – 15

IaaS/PaaS/SaaS Examples – 16

Management Responsibilities of As-A-Service – 17

As-A-Service in Layers – 18

Introduction to Cloud Computing

Infrastructure as a Service – 19

Gartner Magic Quadrant 2015 for Cloud IaaS – 19

Public IaaS (AWS/Azure) – 20

Platform as a Service – 20

PaaS: Cloud Examples – 21

Gartner Magic Quadrant 2015 for Cloud PaaS – 21

Public PaaS – 22

Software as a Service – 23

SaaS Examples – 23

As-A-Service – 24

Cloud deployment models – 24

Public: Netflix – 25

Private: Government Departments – 25

Hybrid: Zynga – 26

Community: NYSE Euronext – 26

Security in the Cloud – 27

Security: Database and hardware – 27

Security: Physical Network and Virtualisation – 27

Security: Virtual Network – 28

Security: OS, Applications, Middleware – 28

Security: Authentication and Authorisation – 29

Security: Data – 30

Security: Data Encryption – 30

Next Steps – 31

WHAT IS THE CLOUD?

The terms ‘cloud computing’ and ‘the cloud’ have been used to describe all kinds of different technology.

Are we talking about Distributed computing? Networked Services? Virtualised Servers or Hosted services?

The actual definition of cloud computing as reported by NIST (National Institute of Standards and Technology) is:

- “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”

CHARACTERISTICS OF CLOUD COMPUTING

To break down this definition into its key parts, Cloud Computing is:

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service

The NIST standard also defines:

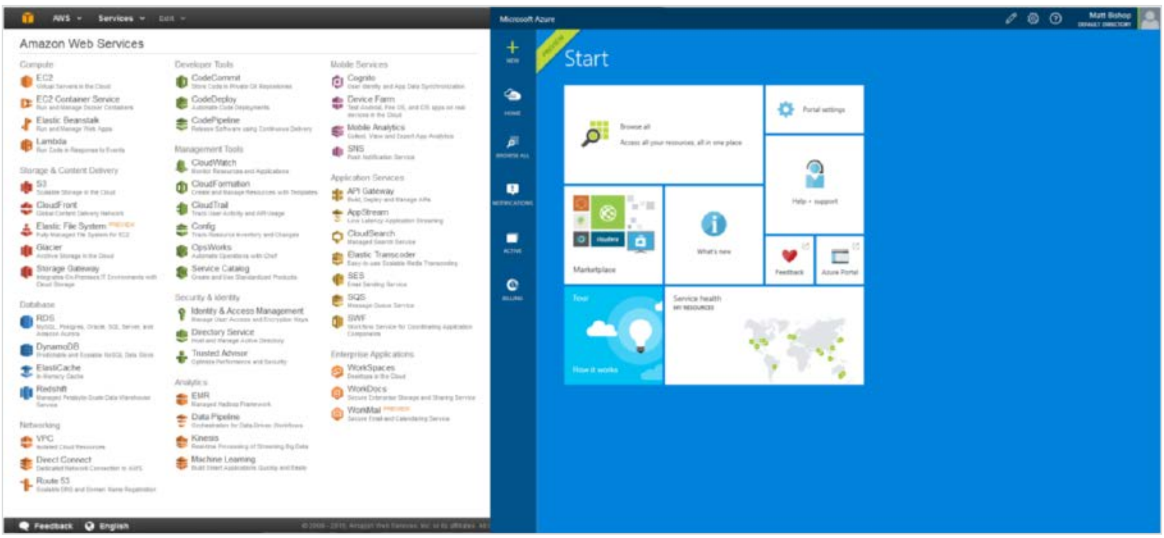
- Three service models (IaaS, PaaS, SaaS)
  - **Infrastructure as a Service**
    - Highly automated and scalable compute resources, complemented by cloud storage and network capability
  - **Platforms as a Service**
    - Provides a platform on which software can be developed and deployed – e.g. Heroku, Google App Engine, RedHat’s OpenShift

- **Software as a Service**
  - Clients (web browsers) provide point of access to software running on servers – e.g. Google Apps, DropBox, etc
- **Four deployment models**
  - **public** – cloud services are delivered over a network is open for public usage e.g. Google
  - **private** – (aka internal cloud) platform is implemented on a cloud-based secure environment, safeguarded by a firewall
  - **hybrid** – 2 or more servers (private/public) that are bound together but remain individual entities
  - **community** – setup is mutually shared between many organisations that belong to a particular community (banks or trading firms)

ON-DEMAND SELF-SERVICE

The customer can separately provision computing capabilities.

Usually via some kind of dashboard e.g. Amazon Web Services, Windows Azure.



BROAD NETWORK ACCESS

Access over network via ‘standard mechanisms’ which would generally be taken to mean standard protocols like HTTP or TCP.

Services should be accessible to a variety of clients running on various hardware (phones, laptops, desktops).

In other words, if it’s only accessible using a proprietary protocol or data format, from custom client, it’s probably not cloud computing.

Notice that ‘network-accessible’ is not the same thing as ‘internet-accessible’; there is no such thing as a private cloud on a public network.

RESOURCE POOLING

Cloud services are provided to multiple tenants (users, applications) by a pool of interchangeable resources. If each tenant needs its own, specific, customised resources, then it’s not cloud computing. Providing on-demand resources with utility pricing can only make economic sense if the resources come from a shared pool.

These resources are dynamically assigned and resassigned in order to get optimal use of out of them.

- Storage, processing, memory, etc

Location independent

- Customer generally does not know/need to know exact physical location of the resources
- For regulatory and architectural reasons, the customer is generally able to specify a general location (e.g. country)
  - e.g. can this data be stored outside of the EU
  - or, can the application function well if the webserver is in the EU and the data it uses is in Australia

RAPID ELASTICITY

Elasticity is a fundamental property of the cloud.

The ability to use exactly the resources you need, without either under-provisioning or excessively over-provisioning, is one of the key benefits of cloud services.

Allows the customer to scale in and out with demand.

- Note: usually scaling in and out, as opposed to scaling up and down

May even be automatic or transparent to the customer.

MEASURED SERVICE

If resources are being dynamically provisioned, it’s essential that the customer should be able to monitor the performance and usage of those resources in real time.

Most cloud resources are offered on a pay-per-use basis, and the customer must be able to monitor their usage in order to control their costs.

BENEFITS OF THE CLOUD

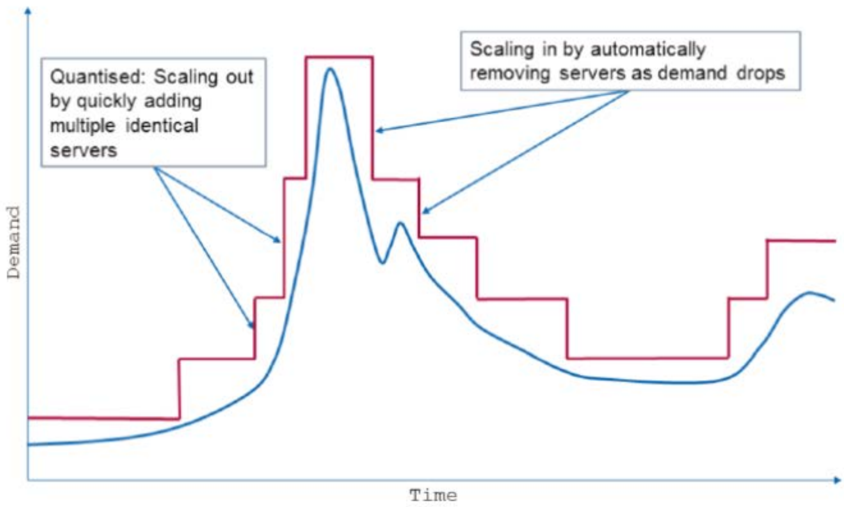
OPEN DISCUSSION

What appeals to you about cloud services and cloud computing?

What doesn’t? What makes you wary or unsure?

Who has used cloud services before?

CLOUD MODEL OF IT PROVISIONING



CLOUD BENEFITS

Over the next series of slides we will be looking at a variety of key components within cloud computing.

- Scale Out
  - Cope with spikes in demand (expected and unexpected)
  - Scale out from zero, for batch processing
- Scale In
  - In response to lower demand
  - Automatically based on time of day (peak hours)
  - Lower costs

Another major benefit is allowing customers to take advantage of the competencies of the cloud service provider.

- Operating a Data centre is complex and many costs and issues need to be considered:
  - Security, certifications, audits
  - Hardware specifications, warranties, installation, disposal
  - Operating Systems and platform licenses

- › Network configuration and security
- › Software has to be self managed

Using a cloud based solution enables us to have specialised teams for all of the above

- Software can be run as a Service
- Infrastructure can be managed and maintained for you
- Allows you to focus on the core competencies of the business

HIGH AVAILABILITY

Running an application in a variety of physical locations was previous so expensive that only the largest organisation could really consider it, and usually only for the most critical of functionality.

With the cloud, the idea of High Availability is essentially free:

- Same cost to deploy to multiple data centres
- Fewer single points of failure
- Lower latency, high availability

AGILITY

Using the cloud tends to increase agility. We'll see many case studies, later on in this course, where faster time-to-market was one of the major benefits of an organisation's move to the cloud.

Faster provisioning for (e.g.) new projects

Faster introduction of new services

- In DevOps for example; agile creation, testing and deploying of services
- Continuous Delivery/Deployment

CLOUD-ENABLING TECHNOLOGIES

FAST, WIDE AREA NETWORKS

In order to get the location-independence of the cloud, you need a fast wide-area network. Otherwise the users and the servers all need to be physically close, or connected by a dedicated network.

With a fast, wide-area network, we can support users in many different locations, and we can support high-availability and disaster recovery solutions that would previously have been unworkable.

POWERFUL, INEXPENSIVE SERVER COMPUTERS

In order to get the pool of interchangeable resources required by a cloud, we need to be able to buy lots of servers and configurable them, and they need to be reasonably powerful.

Building a data centre is a specialist skill, and there are standards and bodies of knowledge related to it.

Data centres are becoming increasingly efficient and cheap, to the extent that it is possible to get a containerised 'modular data centre' shipped to you.

When choosing a cloud provider, they will probably be able to document the standards to which their data centre adheres.

TIA-942 and Uptime Institute are the two best-known examples.

HIGH-PERFORMANCE VIRTUALISATION

In order to get the location-independence of the cloud, you need a fast, wide-area network.

Otherwise the users a.

With a fast, wide-area network, we can support users in many different locations, and we can support high-availability and disaster recover solutions that would previously have been unworkable.

BENEFITS OF VIRTUALISTION

Benefits can include:

- Elasticity/scalability
- Resource pooling on common infrastructure
- More efficient use of physical resources
- Granularity of monitoring and pricing

The hypervisor should prevent issues like:

- Security concerns – leakage of data from one guest to another

'Noisy neighbours' – one guest using all the bandwidth/CPU/etcnd the servers all need to be physically close, or connected by a dedicated network .

VIRTUALISATION VS MULTI-TENANTING

Virtualisation is a type of multi-tenanting, in that it allows multiple users to share a single physical resource. It does this by separating the users at the hypervisor level.

- Namely, every customer has their own virtual machine, from the OS up

However, when we use the term multi-tenanting, we usually use it to refer to application-level separation of users within a single virtual machine.

- The virtual machine is hared by many customers
- Isolation is enforced by the application

Software-as-a-service is frequently multi-tenant, for Cloud services such as Gmail

- "My webmail provider almost certainly uses virtualisation, but when I log on, I don't get a virtual machine all to myself. Instead, the application prevents me from seeing other users' email"

CONTAINERS

Subdivide a single virtual machine.

Less overhead than creating multiple VMs

- Does not require hardware support or emulation
- Containers are isolated at the user (as opposed to kernel) level

Docker is the best-known container technology (docker.com).

Package an application and its dependencies as a unit.

- Run on any Linux server: flexible and portable
- Windows support expected soon
- Integrated into many other cloud products

SERVICES

- Service-Oriented Architectures
- Microservices
- Service-Oriented Cloud Computing Infrastructure
- X-as-a-Service (IaaS, PaaS & SaaS)

SERVICE-ORIENTED ARCHITECTURE

Service: Self-contained, loose-coupled unit of functionality

Each service should do one single thing

- Public contract (interface) but private implementation details
- Abstract, stateless, discoverable, composable
- Get all the usual benefits of reusable modular components

Idea is to create applications by combining services

- Easier to develop, debug and maintain
  - › Can reuse existing components
- Easier to be resilient to service failures
  - › If one service fails, the rest of the app can carry on
- Easier to swap out one service for another
  - › e.g. switching payment processor or mapping provider

SOA AND THE CLOUD

A classic problem with SOA is 'what's in it for me?'

Creating robust reusable services is more difficult than creating something for one-time use.

- Requires work to define that public interface
- Must be made available to other users somehow
- Suspicion of over-engineering – YAGNI ('Ya ain't Gonna Need It')

In the context of a cloud strategy, SOA can make more sense

- Other applications will be in the same environment
- More likely to be able to reuse a service
- Possibility of monetisation
- Particularly to SaaS (Software as a Service) environments

MICROSERVICES

Service-orientation taken to an extreme

Many small, highly-decoupled services, each with a single task

Very different from a tiered/layered architecture

- May be running multiple versions of the same service
- May require service discovery (e.g. Zookeeper, Eureka)

Slightly different from a service-oriented architecture

- Services might not be reusable across applications
- Not really about integration and reuse
- More focused on modularity, being replaceable
- Optimised for continuous delivery/DevOps
- Good fit for containerisation via e.g. Docker

MICROSERVICES EXAMPLE: AMAZON HOMEPAGE

Not monolithic – actually built from hundreds of reusable services.

Amazon use a DevOps processes, and deploy over 50 million times a year (about once in every 1.5 seconds). This sort of DevOps approach can probably only work in a cloud environment.

Amazon have a policy of ‘two-pizza teams’: the team that creates a service should contain very few people – no more than can be fed with two pizzas.

SOCCI: SERVICE-ORIENTED CLOUD COMPUTING INFRASTRUCTURE

Now we have access to virtualisation and resource pooling.

So service-orientation can now be applied to infrastructure.

- “SOCCI can be defined as a service-oriented, utility-based, manageable, scalable on-demand infrastructure that supports essential cloud characteristics, service and deployment models“

This gives us what we need in order to implement and manage an infrastructure-as-a-Service environment.

- SOCCI is the underlying infrastructure for IaaS

More about SOCCI:

- *http://www.opengroup.org/soa/source-book-socci-index.htm*

IAAS, PAAS AND SAAS

IaaS – Infrastructure as a Service

- Consumer can provision computing, storage and network capacity
- Deploy and run arbitrary applications including operating systems

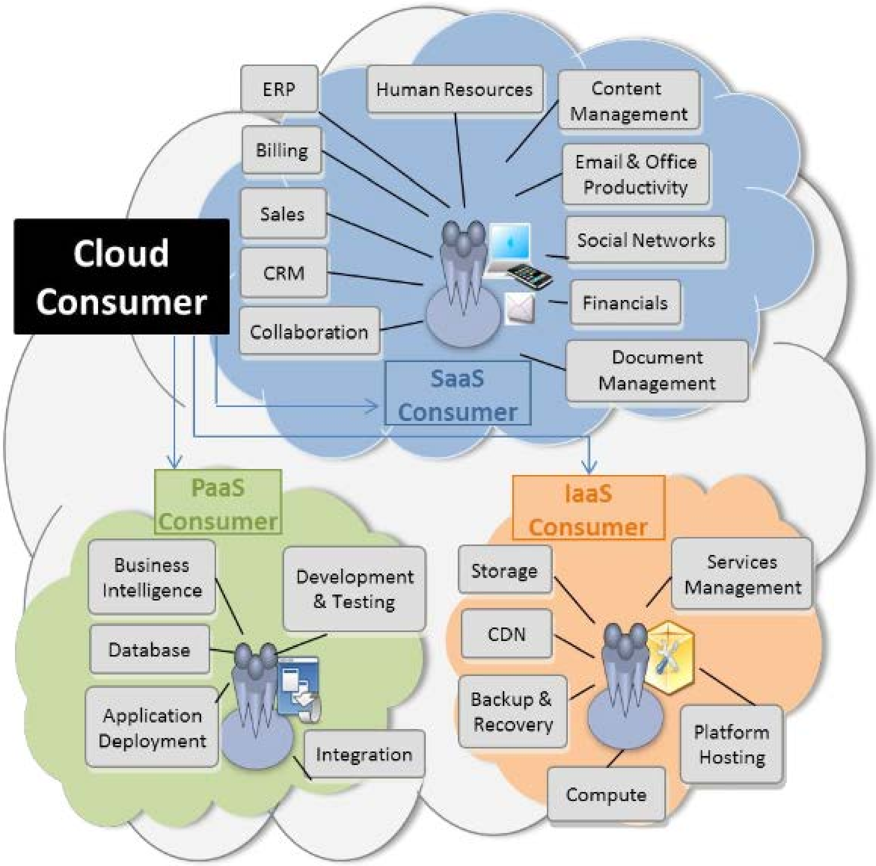
PaaS – Platform as a Service

- Typically built on top of IaaS
- Consumer deploys custom applications, but has limited or no access to the hosting environment, operating system, etc

SaaS – Software as a Service

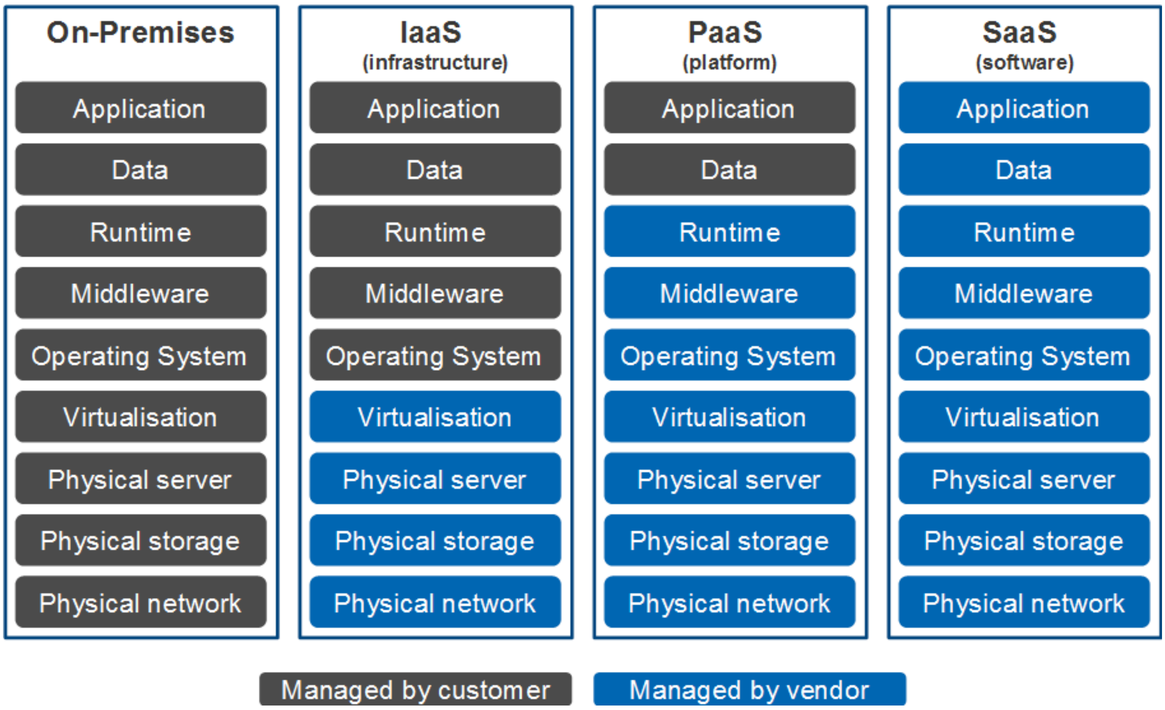
- Typically built on top of PaaS
- Consumer uses existing applications hosted in the cloud
- Consumer may have access to user-specific configuration settings but not to any other configuration or deployment capabilities

IAAS/PAAS/SAAS EXAMPLES



The graphic above shows some of the sorts of functionality that can be provided by IaaS, Paas, and SaaS services.

MANAGEMENT RESPONSIBILITIES OF AS-A-SERVICE



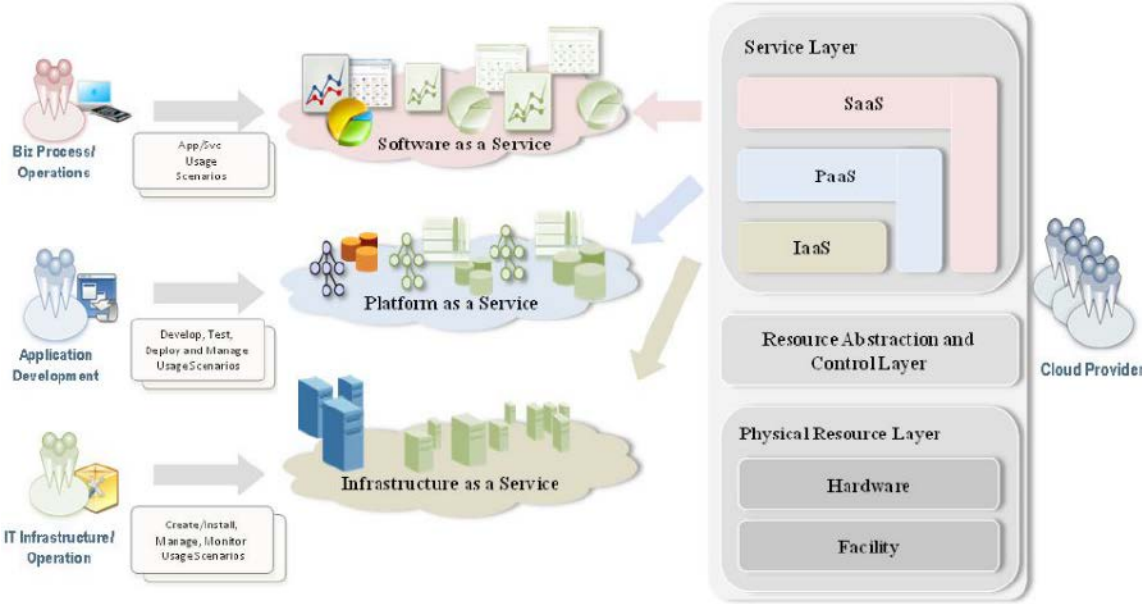
One of the key distinctions between the three service models is in the shared management responsibilities for the infrastructure stack.

IaaS usually divides at the hypervisor, with the provider responsible for everything from the hypervisor downwards, and the consumer responsible for the OS and upwards.

With PaaS, the 'platform' that is provided would include the middleware and runtime (e.g. web server, Java, .NET), all managed by the provider. The consumer need only deploy their own custom application and its data.

With SaaS, the entire stack belongs to the provider, and the customer simply uses it.

AS-A-SERVICE IN LAYERS



The three service models sit in layers, each on top of the next.

Notice in the diagram the ‘resource abstraction and control layer’ ; this is the layer that was referred to in the earlier SOCCI section, which allows providers to manage the underlying hardware.

INFRASTRUCTURE AS A SERVICE

Targeted at IT Operations

Provision infrastructure in the cloud

- Compute, storage, networking, database

Access via specialised dashboard/scripts

Have root/administrator level of control of virtual machines

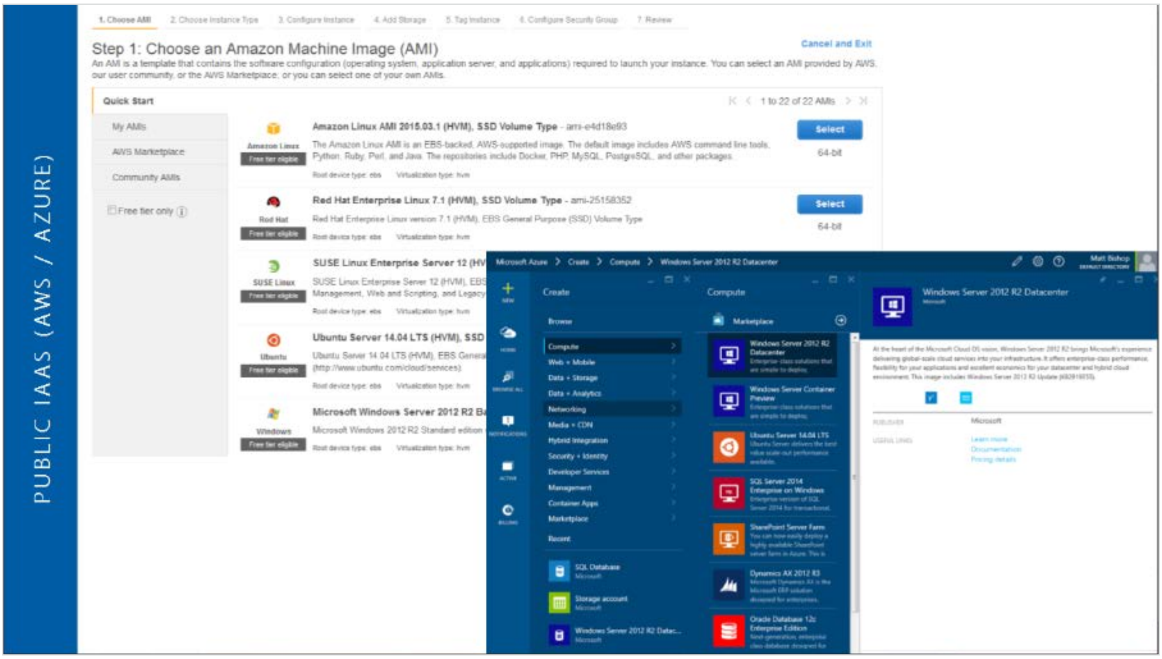
- Install OS, mount disks, create filesystems, install software, etc

No access to physical resources/hypervisor

GARTNER MAGIC QUADRANT 2015 FOR CLOUD IAAS

On the graph, the x-axis is 'completeness of vision' and the Y-axis is 'ability to execute'; being further to the top right is better. The graph clearly shows how dominant Amazon Web Services currently is over all other public cloud providers

PUBLIC IAAS (AWS/AZURE)



PLATFORM AS A SERVICE

Targeted at Developers

Generally the least mature of the three aaS options

Deploy applications to a preconfigured environment

- Easier but less flexible that IaaS
- Closed environment, so consider possible lock-in:
  - ...of code (because coding to a provider-specific API)
  - ...of data (can always dump data to a CSV for export)
- Different languages supported on each platfrom
- Environment often running on another provider’s IaaS
  - e.g. Heroku and OpenShift run on AWS

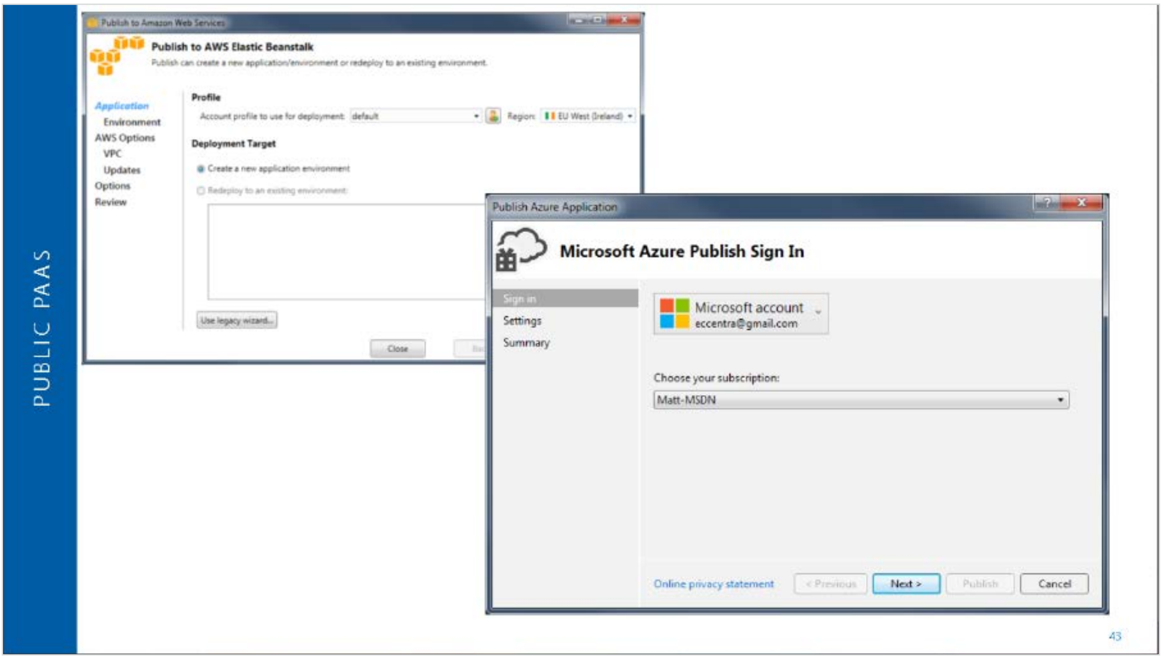
PAAS: CLOUD EXAMPLES

- Cloud Foundry
- Heroku
- Google App Engine
- Engine Yard
- OpenShift Online
- Microsoft Azure
- Amazon Elastic Beanstalk
- Also customisations of existing SaaS applications:
  - force.com (Salesforce)
  - SAP Hana Cloud Platform (SAP Hana)
  - Office 365 (e.g. SharePoint customisations)

GARTNER MAGIC QUADRANT 2015 FOR CLOUD PAAS

On the graph, the x-axis is 'completeness of vision' and the Y-axis is 'ability to execute'; being further to the top right is better. The graph clearly shows the wide range of competing providers

PUBLIC PAAS



PaaS can be integrated with a development environment. On the image above is the AWS Toolkit for the Visual Studio (left) and the Windows Azure SDK for Visual Studio (right), both in the process of deploying an application to their respective clouds.

AWS Elastic Beanstalk is technically IaaS, since it is possible to connect to your new virtual machine as an administrator to install and configure more software. But it behaves like PaaS from a developer's perspective.

SOFTWARE AS A SERVICE

This is the service model that users are most likely to have encountered before; any heavy internet user can probably come up with a lost of dozen SaaS providers without pausing to draw breath – it targets Users or Business Operations.

It is accessed via web browser (or mobile/desktop app).

As seen before, SaaS often sits on a layer of PaaS or IaaS, often from a different provider.

Third party apps running on cloud infrastructure

- Managed and configured by service provider
- Usually free or with per-user subscription model
- Usually built on IaaS or PaaS services
    - For instance Dropbox (a cloud storage tool that synchronizes a folder across multiple machines of all major operating systems, including desktops, tablets and phones) is built on AWS' IaaS object storage (Simple Storage Service, better known as S3)

SAAS EXAMPLES

- Facebook
- Netflix
- Office 365
- Salesforce
- Yahoo!
- OneDrive
- Youtube
- Dropbox
- Gmail
- Twitter

AS-A-SERVICE

It is possible to extend and subdivide the IaaS/PaaS/SaaS model.

The other most common acronyms are:

- SaaS (Storage)
- MaaS (Monitoring)
- NaaS (Network)
- CaaS (Communications)
- DaaS (either Data or Desktop)
- BpaaS (Business Process)
- XaaS (anything/everything)

CLOUD DEPLOYMENT MODELS

Public cloud

- Infrastructure for open use by the general public, on the premise of a cloud provider.

Private cloud

- For exclusive use of a single organisation with multiple consumers (business units). May be on or off premises

Community cloud

- For the use of a specific community of consumer from various organisations with shared concerns (e.g. financial traders, airlines)

Hybrid cloud

- Two or more of the above infrastructures, with some mechanism – standardised or proprietary – for data and application portability between them.

Note that ownership and operation of the infrastructure is unspecified. A private cloud need not be owned by its user.

PUBLIC: NETFLIX

Video streaming at a huge scale

- Up to 1/3 of North American internet traffic
- Started out with Amazon Web Service (IaaS)
  - “We want to use clouds, not build them”
- Built their own PaaS services on top of the AWS
  - Attempt to isolate developers from the AWS API
  - Portability as a long-term goal
- Trailblazer/reference architecture
- Many OSS projects: Simian Army, Asgard, ICE

PRIVATE: GOVERNMENT DEPARTMENTS

Maintenance of own private servers and infrastructure

- Necessary to host government files on private servers
- Maximises security for the departments

Examples include:

- Home Office
- Ministry of Defence
- Department for Work & Pensions

HYBRID: ZYNGA

Social gaming, with >250m monthly active users.

Traditional infrastructures until 2009, then...

- "We couldn't get power fast enough. We couldn't get servers fast enough. We just couldn't scale our infrastructure to match the needs of Farmville"

Moved to AWS to accommodate rapid growth.

Now 80% private cloud (zCloud), 20% AWS

- zCloud is built on AWS principles, using CloudStack
    - Customised to specific requirements of Zynga's workload
- Both clouds managed together, using RightScale

COMMUNITY: NYSE EURONEXT

Global operator of financial markets and provider of trading technologies

Industry needs to share data and make trades

- Very expensive to build private servers
- Regulatory and security issues with public cloud
- Extreme sensitivity to latency
- Not internet-connected at all

Capital Markets Community Platform

- VMWare-based community cloud, offering IaaS to the industry

SECURITY IN THE CLOUD
SECURITY: DATABASE AND HARDWARE

Cloud provider should be able to document:

- Datacentre access
- Hardware access (logical and physical)
- Destruction of failed hardware
- Audits and certifications (ISO 27001, SOC, ITAR, FIPS, PCI, etc)
    - Compliance usually requires audits of both provider and customer

SECURITY: PHYSICAL NETWORK AND VIRTUALISATION

Attacks from outside the cloud:

- DDOS attacks
- Attacks on API Endpoints

Attacks from inside the cloud:

- IP spoofing
- Packet sniffing
- Port scanning
- Zeroing data in newly-assigned RAM or disk
- Other malicious behaviour

These can be prevented or mitigated by the cloud provider:

- Standard and proprietary networking techniques
- Enforcement in the hypervisor
- Terms of Service

SECURITY: VIRTUAL NETWORK

For IaaS, must be able to configure:

- Firewalls (allowed and denied ports and protocols)
- Virtual software (software-defined) networking
  - Subnets (public/private)
  - IP addresses (public/private/persistent across reboots)
  - Routing tables
  - Access control lists
  - Gateways to internet and/or VPN
  - NAT

Software defences:

- Bastion hosts
- Intrusion detection/prevention systems
- Anti-malware, anti-virus, etc

SECURITY: OS, APPLICATIONS, MIDDLEWARE

These need to be protected and patched

- Exactly as for a traditional on-premises deployment

Whose responsibility is that?

- It depends on the service model

In IaaS, these are the customer's responsibility

In PaaS and SaaS, they're the provider's responsibility

SECURITY: AUTHENTICATION AND AUTHORISATION

Who can access cloud service provider?

- e.g. in IaaS, to provision new servers and logical infrastructure
- Also manage account billing, etc
- Not just dashboard – must secure API as well!

There are various layers that typically all have their own authentication mechanisms: the provider's own APIs, the operating system, the installed applications.

These might be three entirely different groups of users. For instance, in AWS:

- The first is me, logging in with my email address and password, plus any users I choose to create in the AWS console (who log in at a special URL I give them)
- The second is up to me; I have the SSH keypair, and can give it to anyone I want. AWS does not control, or know about, those users. If I want to, I can create new users in my OS, or (for Windows) install Active Directory and create users there; it's all up to me
- The third depends on my application. It might be my Active Directory users, or I might let any internet user create an account

For the first, then, my cloud provider (in this case AWS) should help me manage the users and assign them permissions. For the second and third, my cloud provider has nothing to do with it, and it's up to me.

SECURITY: DATA

Where is data stored and/or transmitted?

- Regulatory issues may require e.g. only within EU
- e.g. Amazon EU-West-1 region physically in Dublin
  - Multiple physical Dublin datacentres enable HA
  - But there are a few US-only API endpoints (e.g. Simple Email Service)

For IaaS, customer can control this to some extent.

.

For SaaS, is generally imposed by provider

- Require careful reading of ToS, privacy policy, applicable laws
- End user may not even realise their data is in the cloud
  - Where, physically, is my instagram/Vine/Dropbox data?
  - All of them are in Amazon's US datacentres

SECURITY: DATA ENCRYPTION

Is data encrypted at rest? How about in motion?

SSL/TLS support

Support for encryption:

- Transparently as part of a service (e.g. S3)
- In applications (e.g. on writing to a database in a PaaS solution)
- In filesystems (e.g. when using a volume in an IaaS solution)

May present key management issues

- Perhaps solvable using tools like CloudHSM

NEXT STEPS

Explore some of the cloud tech available

- AWS
- Azure
- Google Cloud