

Chapter XX

Cloud Computing

Amazon Web Services (AWS)

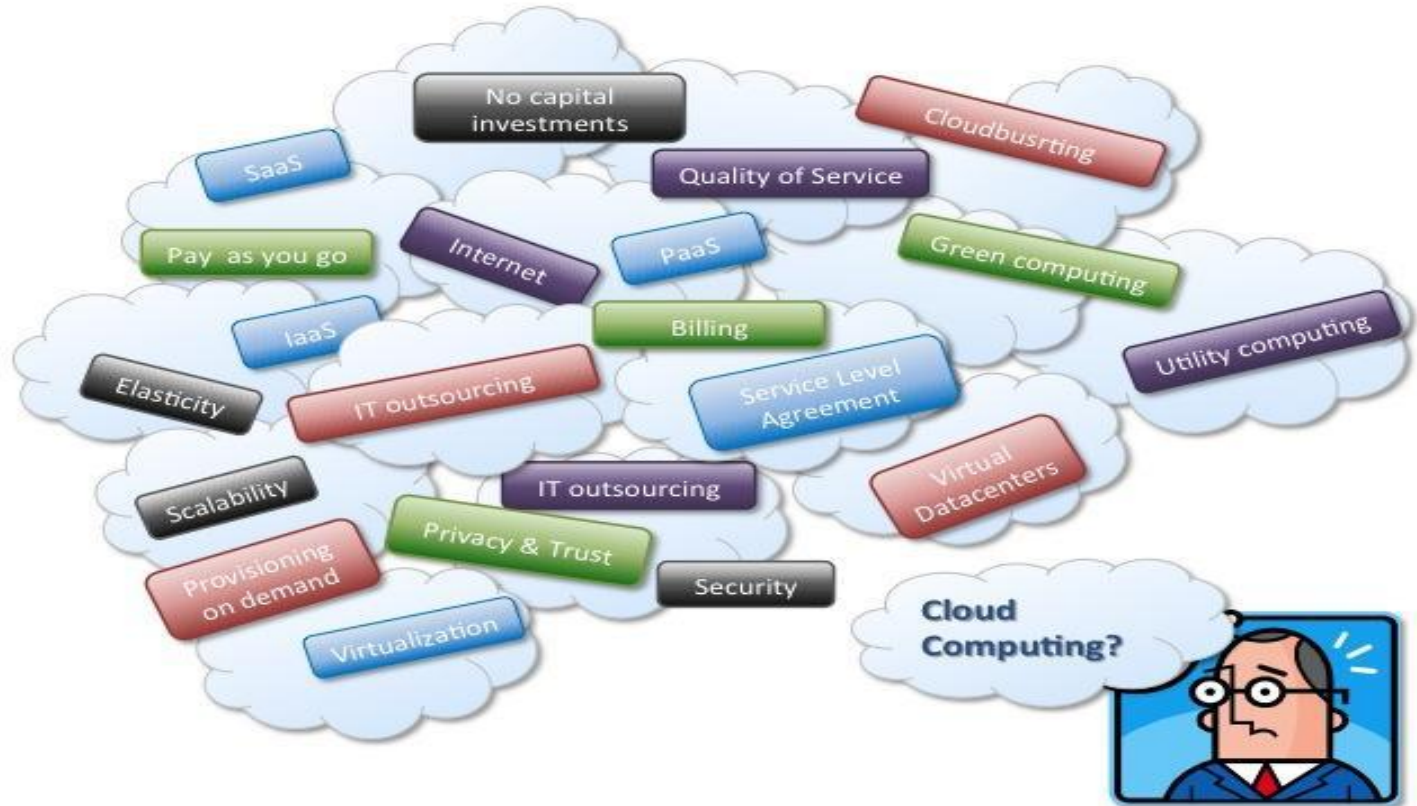
A decorative graphic in the top right corner consisting of a circular cluster of various blue and white technology icons (like a smartphone, laptop, gear, and person) with several network cables (Ethernet and USB) extending from the bottom right towards the center.

Dr. Paul Talaga

As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of 'computer utilities' which, like present electric telephone utilities, will service individual homes and offices across the country.

-Leonard Kleinrock, 1969, ARPANET

Defining Cloud Computing



Huh?

- It's a Buzzword!
- Term for MANY ideas and concepts

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and software in the datacenters that provide those services. - Armburst

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. - NIST

What types of clouds exist?

Cloud Deployment Models

Public/Internet Clouds

- * 3rd party, multi-tenant Cloud infrastructure & services:
- * available on subscription basis to all.



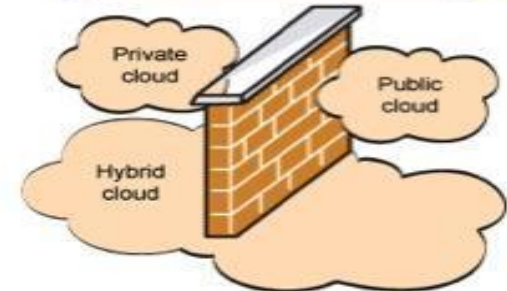
Private/Enterprise Clouds

- * A public Cloud model within a company's own Data Center / infrastructure for internal and/or partners use.

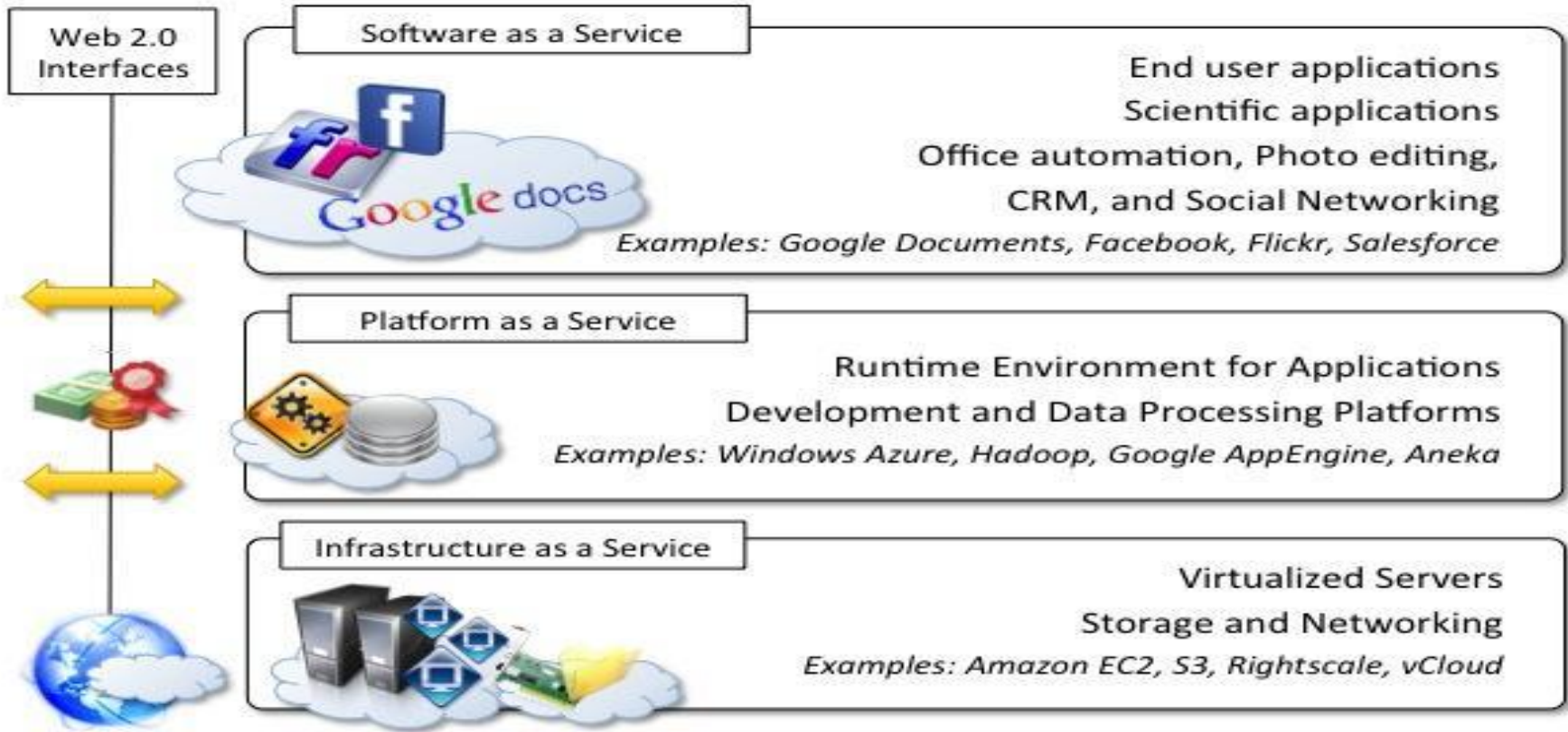


Hybrid/Inter Clouds

- * Mixed usage of private and public Clouds: Leasing public cloud services when private cloud capacity is insufficient



Reference Models



What are the benefits to a (regular) business?

- No up-front commitments
- On-demand access
- Nice pricing (capital costs -> utility costs, no depreciation)
- Simplified app acceleration and scalability
- Efficient resource allocation
- Energy efficiency??
- Seamless creation and use of third-party services

Benefits for a software company going to SaaS?

- NO deployment issues (CDs, downloads, etc)
- No need to support multiple OSs
- Faster to market
- A/B testing of features
- Efficiency and reliability now key
- More efficient developers, just try it!

Current Platforms

- Amazon Web Services (AWS) - Current leader in IaaS: Elastic Compute Cloud (EC2), Simple Storage Service (S3), many many others.....
- Google AppEngine - PaaS - Python, Java, Go
- MS Azure - IaaS & PaaS
- Hadoop - Open Source framework for MapReduce
- Force.com & Salesforce.com

AWS History (from wikipedia)

Isn't Amazon an online retailer?

Yes! And they need to manage lots of computers!

Sell their internally developed technology!

- 2003/2004 - Paper on idea - Simple Queue Service - EC2 (Cape Town)
- 2006 - Full AWS launch - SOAP/REST
- 2010 - All amazon.com retail on AWS
- Outages - 2011, 2012, every year?

AWS Ecosystem



Compute Services

Elastic Compute Cloud (EC2)

- Virtual machines - IaaS
- User can select:
 - Amazon Machine Image (AMI) - template to start a VM (market, or create your own)
 - # cores - ECU (EC2 Compute Unit) - 1 ECU = 2007 Opteron/Xeon
 - Memory
 - Local storage
 - Network & Security (firewall specifications)
 - Location and availability zone
 - # to start!

EC2 Details....

- How to access securely? Passwordless SSH (private/public keys) for Linux
 - In unix, `ssh-keygen` to make pair
 - Can use your github public key!
 - Your public key gets injected into the image
 - `ec2-user` for username, unless noted
- Windows must use an Admin password retrieval system. Needs your private key.
- Must define a security policy: what TCP ports to allow in. 22 (ssh), 80 (http?) *and from where.*

EC2 Instance Types

What type of machine to start:

General purpose - M1, ...M5, M6 Arm Processors - A1 Burstable T2, T3...

small | medium | large | xlarge

Compute optimized - C1,...C4, C5...

small | medium | large | xlarge

Storage optimized - I3, D2, D3, H1

small | medium | large | xlarge

Memory Performance - R4, R5, R6, X1.. z1d

Large, xlarge, 2xlarge, 3xlarge, 6xlarge, 12xlarge, metal

Accelerated (NVIDIA A100, Gaudi, NVIDIA M60, FPGA, H.264)- P2, P3, P4, DL1, G3, F1, VT1

Large, xlarge, 3xlarge, 6xlarge, 24xlarge

<http://aws.amazon.com/ec2/instance-types/>

What type of OS (AMI)?

- [awsMarketplace](#) - buy/sell AMI's!
 - Some AMI's charge extra /hr
- Pre-configured software!
 - MongoDB, Wordpress, Ruby Stack
- Linux Free/Paid
 - Amazon Linux, Debian, RedHat, SUSE
- Windows
 - Server 2008, 2012, 2019
- Or create your own!!!

Pricing!!!!

- New users get access to FREE TIER, for a year.
- Cost for running EC2 /second* (rounded up)
 - On-demand, reserve, spot pricing
- Cost for data transfer
 - Free inbound from internet
 - Tiered outbound to internet
 - Some cost between EC2 instances
 - Cost for data storage (S3, EBS)

Price sheet: <http://aws.amazon.com/ec2/pricing/>

AWS Locations & Availability Zones

- 21 Regions AWS [locations](#) (+ GovCloud)
 - Tokyo, Singapore, Sydney, Sao Paulo, Ireland, N.California, Oregon, N.Virginia, Frankfurt
- CloudFront/Route 53 distributed around globe
- Availability Zones are isolated parts of a region (datacenter)
 - Your zone #'s don't match others!
 - Distribute instances for reliability

View [AWS Status](#)

Useful Notes:

- `ssh -i ~/.ssh/labsuser.pem ec2-user@54.91.120.107`
- **Use** `sudo <command>` **for commands requiring root privileges**
- `sudo yum [search|install] <package>` **as in** `sudo yum install httpd` **for Apache HTTPD Webserver**
- **Manually start a service with** `sudo systemctl start httpd`
`sudo systemctl enable httpd`
- HTTPD default serving location is `/var/www/html/` and only root has write access.
- View HTTPD logs in `/var/log/httpd/*` `access_log` `error_log` but you'll need to be root to view these files.
- `tail <file>` is useful for viewing the end of log files. `tail -f <file>` will follow changes so you can see them in real time.
- If HTTPD isn't working, your firewall `iptables` may be running. `sudo iptables -L -n` to see config. `sudo service iptables stop` to turn off. (Security risk, but OK for our purposes)
- `wget http://someURLhere/somepage.html` - Download a page in terminal
- Can run startup code using User data `base64 encoded`:
 `#!`
 `touch ~/PLEASE_WORK.txt`

Load Balancers:

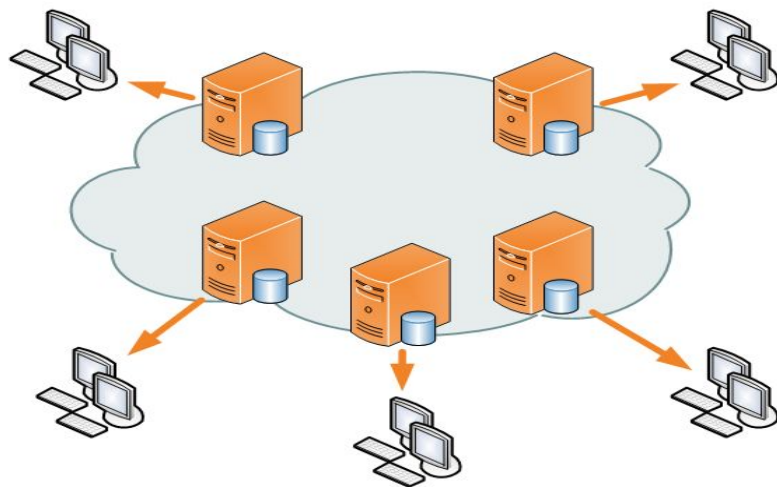
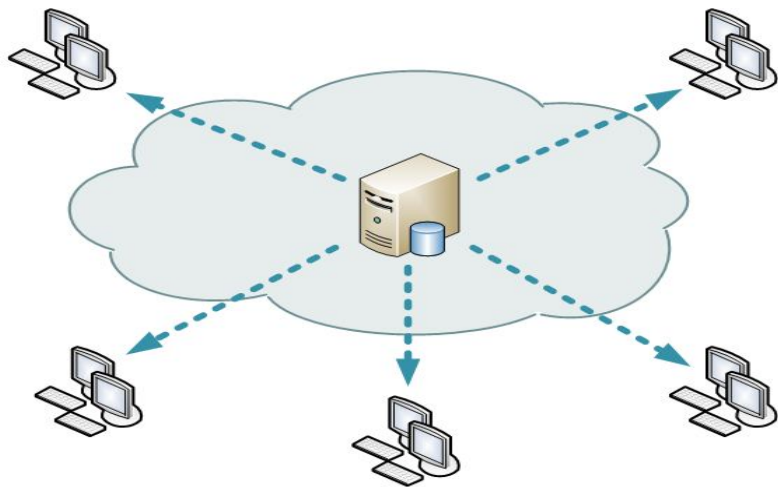
- Allows requests to be distributed over many services
- Allows HTTPS 'unwrapping'
- Requires

AWS EC2 API

- Allows *programmatic* control over all Dashboard functions!
- Allows programs to monitor and control infrastructure!
- Can minimize costs
- See me for more details

AWS Cloudfront

- Amazon's Content Delivery Network (CDN) - similar to [Akamai](#), [Limelight](#), and [others](#)



Simple Storage Service (S3)

- REST interface
- Eventually consistent!
- Cost: \$0.03 / GB / Month + transfer
- 2 level hierarchy
 - Buckets - ALL users share namespace!
 - Objects - 'files' in a bucket key:value
- Accessible: http://s3.amazonaws.com/bucket_name
renaming possible via download, delete, upload.
- Access:
 - S3 Management Console
 - BOTO [API](#) - [Examples](#)

AWS Glacier

- AWS Long term storage (backups)
- INFREQUENT access - hours to retrieve
- Cheap! (S3 - \$0.03->\$0.027 / GB / Month)
 - Storage: \$ 0.004 / GB / month (VA)
 - Download: 5% total /month free, then \$0.05/GB
 - Transfer: (out of AWS) 1st GB free
- Access:
 - Web Dashboard
 - BOTO API - S3 - Bucket Lifecycle Policy
 - BOTO API - Glacier

AWS Lambda

- On-demand code execution service
- Auto-scaling as needed
- Interpreted languages supported
 - Like [Google Cloud Functions](#) and [Cloud Run](#)
- This runs Alexa!
- You pay for the RAM and ms you use.
- The basis for a 'serverless' web app. [Tutorial](#)
- Why run a server constantly if you only need it periodically?

AWS Serverless Computing

- Serve static HTML & CSS
 - CDN, Load Balancer, AWS Amplify ([link](#))
- Site (users) interact with app through AJAX
- AJAX calls [API Gateway](#) -> AWS Lambda
- AWS Lambda does what it needs
 - Do computation, update a [DB](#), make decisions
- AJAX request satisfied and 'page' updated