**What is Cloud computing**

* Infinite computing. Near infinite compute, CPU storate, CPU memory.
* Elimination of upfront cost for infrastructure building.
* Utility – if you use resources inefficiently, you pay a lot of money, but if you can use efficiently, you can treat it like a utility.
* Comparative advantage – you focus on things you are best at and leave other things to other people.

**Week 1**

**[1.0] [5 points] The evolution of Cloud Computing has been compared to the evolution of electricity supply as a utility. Describe specific problems that Cloud Computing solves as compared to businesses running their own data centres.**

When businesses run their own data canters, they face the following problems:

* Need up-front payment – with cloud you solved with pay as you go. Pay as you go model is cost-effective, so you pay only for resources that you use.
* Difficult to scale and it is expensive - with cloud you have instant access to any region
* Availability, reliability – if network goes down, service will suffer- with cloud you have guarantee from the provider for exactly how many minutes in a year you will have downtime.

**[2.0] [5 points] Describe the different categories of services (XaaS) cloud computing can provide with specific examples of each service.**

The most common types are Saas, Paas, Iaas, however, certain cervices within these categories can be offered such as Dass, Caas.

Saas: Software as a service is provided on substription basis and is centrally hosted. Email, Dropbox, Calendar. Software that you can rent and incorporate into your infrastructure. In short is at a large scale, renting out capabilities.

Gmail, no hosting the web server.

-Splunk, Data Dog

Iaas: Provide all the required physical infrastructure and computing resources on the cloud. AWS, Azure. Is like a big warehouse store where you get things in bulk but you still have to make sure that you can configure all that equipment and tie it together. Get things in bulk and the cost is very low.

Amazon: EC2 A lot like Costco.

You spin up virtual machines, setup the networking layer, but at a significant cost savings.

Paas: Platform as a Service is a high-level abstraction that allows the SW team to focus on the logic only and infrastructure provider will take care of the rest of the work of deploying that into production. You pay bit more.

Full-service gasoline fill up at a station vs doing it yourself.

Ex: Heroku

Google App Engine (GAE)

Amazon: Beanstalk

MaaS(Metal as a Service) is a newer term. A lot of the cloud paradigms that you use can be applied to physical hardware. This is when you want to deal with specialized situations like GPU programming or maybe large storage systems.

The ability to spin up and provision machines yourself. You can physically control servers. GPU – You may have a very specialized multi-GPU setup for, let’s say, machine learning, or specialized database and you want to control that physical hardware.

Serverless (Function as a Service Faas) newer types of service models where you do not need to worry at all about the underlying infrastructure. It has some similarities and some overlap with Paas. AWS Lambda, a function that you put into the Cloud and you can map events to it. A lot of it is around piece of logic you put on Cloud and hook it up to an event. Having a light bulb in garage when you open garage when it is night on. Abstracting business logic into a unit of work and then applying that, the work wherever you need to.

**[3.0] [10 points] An established financial company is about to launch their new banking application. Give 5 reasons why the company should use their own data centre rather than cloud computing.**

- Regulation

- Both HW, SW and personnel available already. There is already capacity available for use in the existing system. Electricity is too cheap to compare to the cost of cloud services.

- Customer requirement (perception of cloud being less secure)

- Internet network is poor. Afraid of having downtime because of the network.

- No expertise in cloud computing

- it is governmental organization, and they cannot reduce headcount when switching from traditional to cloud.

**[4.0] [20 points] Describe the concepts of vertical and horizontal scale. Describe 2 different ways in which you could scale a web application horizontally. Describe a potential architecture to scale the database to handle the scaling out of the web servers.**

Vertical : adding more power or storage space to existing machine or completely replacing machine with more powerful machine. Data sharing is less expensive as it uses reference within single machine. There is hard peak limit, once it is reached it cannot handle the load anymore.

Horizontal: is when we add more computers and use it as a network of computers to handle the load. No limit for peak handling. Redundancy is easy to create so less downtime.

There is no rule of when to choose what, but it is better to decouple the system so components can be chosen to be horizontally or vertically in scaling depending on the need.

Web server overload – Vertical scaling once it reaches its max capacity architecture remains the same but the capacity to handle more clients is increased. Single point of failure as it is on one server. Multiple server

Factoring sessions. Redis server with redundancy

Query expensive = add cache

Database failure – DB multiplication Master & slave

Queries are expensive

Sharding – multiple databases divide alphabetically.

[6.0] [20 points] Describe the steps which you would take on AWS and the decisions that would need to be made to create, configure and run a Virtual Machine Instance.

Create an EC2 instance using awscli

[1] Create a security group

aws ec2 create-security-group --group-name <student number>-sg --description "security group for development environment"

Note: this will use the default VPC (you will learn about this later in the course) – if you want to specify another VPC, you would use --vpc-id vpc-xxxxxxxx

Note the security group id that is created

[2] Authorise inbound traffic for ssh

aws ec2 authorize-security-group-ingress --group-name <student number>-sg --protocol tcp --port 22 --cidr 0.0.0.0/0

[3] Create a key pair that will allow you to ssh to the EC2 instance

aws ec2 create-key-pair --key-name <student number>-key --query 'KeyMaterial' --output text > <student number>-key.pem

To use this key on Linux, copy the file to a directory ~/.ssh and change the permissions to:

chmod 400 <student number>-key.pem

[4] Create the instance and note the instance id

aws ec2 run-instances --image-id ami-d38a4ab1 --security-group-ids <student number>-sg --count 1 --instance-type t2.micro --key-name <student number>-key --query 'Instances[0].InstanceId'

Optional: Add a tag to your Instance

aws ec2 create-tags --resources i-??????? --tags Key=Name,Value=<student number>

// 18.04 ami-176aa375

[5] Get the public IP address

aws ec2 describe-instances --instance-ids i-<instance id from above> --query 'Reservations[0].Instances[0].PublicIpAddress'

[6] Connect to the instance

ssh -i <student number>-key.pem ubuntu@<IP Address>

[7.0] [10 points] Describe EBS and what features it offers.

AWS EBS provides two types of volumes that are SSD-Backed delivering low-latency with highest IOPS and HDD-Backed for streaming sequential access for highest throughput.

The EBS provides maximum data protection as the failure rate is between 0.1 to 0.2%. The user can still take precautions by taking point-in-time snapshots for efficient backup, Archiving, and recovery for procuring new EC2 instances.

AWS Elastic Block Store provides features in various manners such as security, reliability, availability

Consistent with Low latency performance.

Backup facility along with innovating and restore.

Modify scalability (up and down).

Location Flexibility.

Excellent Performance.

Reliable and secure storage.

Scalable and innovative.

Features of AWS EBS

**Features of Elastic Block Store - AWS EBS**

a. High-Performance Database

Amazon Es creates a storage volume and attaches them to the Amazon EC2. After the attachment, the user can create the file system to run a database. AWS EC2 introduces the variety of databases as per the need of the user to maintain the cost and requirement of the user.

For the security purpose, these are placed in the special availability zone so to protect the data from the failure of the single component.

b. Encryption

To remove the need to build and manage a secure key infrastructure AWS EBS encryption offers encryption of EBS data volumes, boot volumes, and snapshots. The encryption occurs on the servers that host EC2 instances.

This providing encryption of data as it moves between EC2 instances and EBS. For the security reasons, Amazon EC2 is integrated with AWS Identity and Access Management which enables access control to the Amazon EBS volumes.

c. Managing Access

Amazon-managed key and the keys created by the user to provide the security to the data EBS encryption enables data at rest security by encrypting your data volumes, boot volumes, and snapshots.

Access management is done with the help of IAM which is a feature of the AWS account offered at no additional charge. The user will charge only for use of other AWS services by your users.

d. Snapshots

For the long-term durability, AWS EBS offers point-in-time snapshots of EBS volumes. They are backed up to the Amazon S3. In Amazon EBS only the blocks that change after the last snapshot save. The user bill only for the charged block.

e. Elastic Volumes

AWS Elastic Volumes allows the user to create storage and attach them to Amazon EC2. Elastic Volume dynamically increases the capacity and tune performances. It works with no downtime or performance impact.

[8.0] [10 points] What is CLI and Boto? What are advantages of using CLI? How does Boto function helps in AWS operation?

Boto is a software development kit (SDK) designed to improve the use of the Python programming language in Amazon Web Services. The Boto project started as a customer-contributed library to help developers build Python-based applications in the cloud, converting application programming interface (API) responses from AWS into Python classes.

A command-line interface (CLI) is a text-based user interface (UI) used to run programs, manage computer files and interact with the computer.

greater control of an OS or application;

faster management of many operating systems;

ability to store scripts to automate regular tasks;

basic command-line interface knowledge to help with troubleshooting, such as network connection issues.

You use the AWS SDK for Python (Boto3) to create, configure, and manage AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3). The SDK provides an object-oriented API as well as low-level access to AWS services.

[8.0] [10 points] Describe what virtualisation is and describe the characteristic attributes of the different types of virtualisation (Language, Operating System and Hardware).

Virtualization is a structure that segregates a computer resource into manifold execution environments and is it often trusted to craft a virtual version of a resource or a device. It also permits to run multiple operating systems on the same system by providing virtualized hardware to the guest operating system.

Common distinction:

(language-based) virtual machines

Instruction set usually does not resemble any existing architecture

Java VM, .Net CLR, many others

Virtual Machine Monitors (VMM) or Hypervisor

instruction set fully or partially taken from a real architecture

Other types of virtualization

Web Server

Virtual hosts using a different hostname, configuration file

Virtual Environments

Java, Python and Ruby amongst others have the ability to configure separate versions of language and libraries

Containers

Like VMs but run in user space and packages binaries and libraries

Docker, Kubernetes, AWS Containers (Docker)

Serverless Environments

Code is executed in response to an event, including HTTP

[9.0] [10 points] Describe what containers are with reference to Docker and discuss their similarities and differences from operating system virtualisation perspective as provided by VirtualBox.

Docker is Originally based on Linux Containers (LXC) but now on runC

In traditional virtualization, a hypervisor virtualizes physical hardware. The result is that each virtual machine contains a guest OS, a virtual copy of the hardware that the OS requires to run and an application and its associated libraries and dependencies. VMs with different operating systems can be run on the same physical server. For example, a VMware VM can run next to a Linux VM, which runs next to a Microsoft VM, etc.

Instead of virtualizing the underlying hardware, containers virtualize the operating system (typically Linux or Windows) so each individual container contains only the application and its libraries and dependencies. Containers are small, fast, and portable because, unlike a virtual machine, containers do not need to include a guest OS in every instance and can, instead, simply leverage the features and resources of the host OS.

Just like virtual machines, containers allow developers to improve CPU and memory utilization of physical machines. Containers go even further, however, because they also enable microservice architectures, where application components can be deployed and scaled more granularly. This is an attractive alternative to having to scale up an entire monolithic application because a single component is struggling with load.

[10.0] [10 points] There was an evolution of operating system during the half century from 1960 to 2010. Identify the virtualisation milestones in this above evolution and explain them briefly.

First VM architected by IBM in 1972 VM/370 to provide full VM of mainframe machine

1997 Virtual PC for Mac by Connectix

1999 VMware’s VMware Virtual Platform

2003 Open Source hypervisor Xen

2005 VMware Player – free VM player

2007 VirtualBox

In the mid-1960s, IBM's Cambridge Scientific Center developed CP-40, the first version of CP/CMS. It went into production use in January 1967.

IBM announced the System/370 in 1970. To the disappointment of CP/CMS users – as with the System/360 announcement – the series would not include virtual memory.

On February 8, 1999, VMware introduced the first x86 virtualization product, VMware Virtual Platform, based on earlier research by its founders at Stanford University.

VMware releases VMware Workstation 6.5 beta, the first program for Windows and Linux to enable DirectX 9 accelerated graphics on Windows XP guests .

Docker, Inc. releases Docker, a series of platform as a service (PaaS) products that use OS-level virtualization.

[10.0] [5 points] You are asked to store data about music albums in a DynamoDB table. For each album, you need to record the title of the album and the artist name. Describe the commands you would use to create a table to store such information and write an entry to that table in DynamoDB.

Create Table:

aws dynamodb create-table --table-name Music

--attribute-definitions \

AttributeName=Artist,AttributeType=S \

AttributeName=SongTitle,AttributeType=S

--key-schema AttributeName=Artist,KeyType=HASH \

AttributeName=SongTitle,KeyType=RANGE \

--provisioned-throughput ReadCapacityUnits=1,WriteCapacityUnits=1 \

--endpoint-url=http://localhost:8000

aws dynamodb scan --table-name Music \

--endpoint-url=http://localhost:8000

Create Entries:

aws dynamodb put-item \

--table-name Music \

--item \ '{"Artist": {"S": "No One You Know"}, "SongTitle": {"S": "Call Me Today"},

"AlbumTitle": {"S": "Somewhat Famous"}}’ \

--return-consumed-capacity TOTAL --endpoint-url=http://localhost:8000

aws dynamodb put-item \

--table-name Music \

--item '{"Artist": {"S": "Acme Band"}, "SongTitle": {"S": "Happy Day"}, \

"AlbumTitle": {"S": "Songs About Life"} }’ \

--return-consumed-capacity TOTAL --endpoint-url=http://localhost:8000

aws dynamodb scan --table-name Music \

--endpoint-url=http://localhost:8000

Query:

{ "Artist": { "AttributeValueList": [{ "S": "No One You Know" } ],

"ComparisonOperator": "EQ"

},

"SongTitle": { "AttributeValueList": [{ "S": "Call Me Today" } ],

"ComparisonOperator": "EQ"

}

}

aws dynamodb query --table-name Music --key-conditions= file://key-conditions.json \

--endpoint-url=http://localhost:8000

[11.0] [5 points] Describe how S3 handles consistency of objects and how this approach affects the state of objects when they are read using a GET.

S3 handles consistency through versioning rather than locking

The idea: every bucket + key maps to a list of versions

[bucket+key]  [object v1] [object v2] [object v3] …

Each time we PUT an object, it gets a new version

The last-received PUT overwrites any previous ones!

When we GET:

An unversioned request likely receives the last version – but this is not guaranteed depending on propagation delays

A request for bucket + key + version uniquely maps to a single object

[12.0] [5 points] What are the core components of DynamoDB

Tables

Collection of data (e.g. representing entities): People

Schemaless

Items

Group of attributes that is uniquely identifiable

Person in People table

Physically a JSON document

Atrributes

Fundamental data element

Limited types of data: Scalar, Document, Sets

LastName in Person item

One or more attributes make a primary key (unique)

[13.0] [5 points] When a Bucket is created, AWS allows the specification of a number of features that can be managed. What are the key properties and features?

Once created can manage

Lifecycle

Transition objects that are infrequently accessed (or after fixed time) to cold storage

Replication

Automatically copy objects to another bucket in a different region

Analytics

Suggest how to manage objects based on access patterns

Metrics

Stats on operations on objects in the bucket

Inventory

Provide a regular snapshot of contents of bucket

[14.0] [5 points] We can leave S3 buckets open to public. Is this suitable for a specific application? Why and why not? Justify your answer

S3 buckets left open to public:

Nov 2017 Contractor exposes personally identifiable data from 50,000 Australians (AMP, UGL, Rabobank)

Nov 2017 Accenture leaked corporate information

Alteryx exposes data on 120 million US households

March 2018 Medical Data of 33,000 patients

When this important personal information is leaked, it may raise some information security as well as personal privacy issues. It can also lead to a crisis of trust for the companies involved.