Week1

Q1

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.

Problems of running on own data centers：

- huge investment in data center, including hardware, software, cooling machines, etc.
- need to hire IT experts to maintain the data center.
- scalability issues. It's diffcult to add or remove computers when the demand changes.

1. Reduced IT costs: Reducing the cost for hiring IT experts and maintaining one's own data center by using the resources of the cloud service provider
2. No up-front investment: pay-as-you-go model
3. Scalability: using cloud computing can quickly scale up/down as demand varies.
4. Security: Regular and automatic system updates to ensure data security.

Virtualization in the cloud: pros and cons

- Gives cloud provider a lot of flexibility
  - Can produce VMs with different capabilities
  - Can migrate VMs if necessary (e.g., for maintenance)
  - Can increase load by overcommitting resources
- Provides security and isolation
  - Programs in one VM cannot influence programs in another
- Convenient for users
  - Complete control over the virtual 'hardware' (can install own operating system own applications, ...)
- But: Performance may be hard to predict
  - Load changes in other VMs on the same physical machine may affect the performance seen by the customer

Q2

Describe the different categories of services (XaaS) cloud computing can provide with specific examples of each service.

SaaS: Software as a service

SaaS is a cloud-based software delivery model in which the cloud provider develops and maintains the entire cloud application software that is available for end users to use over the internet.

Example: Google Workspace, Zoom, Microsoft 365.

PaaS:Platform as a Service

PaaS is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

Example:Windows Azure, Google App Engine, AWS ELASTIC Beanstalk.

PaaS vs SaaS: PaaS is on-demand access to a complete, ready-to-use, cloud-hosted platform for developing, running, maintaining and managing applications. SaaS is on-demand access to ready-to-use, cloud-hosted application software

IaaS: infrastructure as a service

Infrastructure as a service (IaaS) is a type of cloud computing service that offers essential compute, storage, and networking resources on demand, on a pay-as-you-go basis, which means it charges according to the actual use or occupation of resources by users.

Example: AWS, Microsoft Azure, DigitalOcean

Hybrid cloud

For businesses seeking the benefits of both private and public cloud deployment models, a hybrid cloud environment is a good option. By combining the two models, a hybrid cloud model provides a more tailored IT solution that meets specific business requirements.

Pros:

- highly flexible and scalable
- cost effective
- enhanced security Cons:
- communication in network level may have problems as hybrid applied both private and public cloud.

In short:

IAAS : Infrastructure as a service – cloud provides raw computing resources e.g. AWS EC2.

PAAS : Platform as a service – provide middleware/infrastructure level e.g. Google app engine.

SAAS: Software as a service. Provide an entire application for users e.g. Microsoft 365.

Q3

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.

1. System management issue: By employing your own data centre, you can directly control and manage the system by yourself. This can get a better understanding of how your system works, and managing the data processing progresses, which is important to a financial company.

2. Security issue: using your own data center is less likely to be hacked or exploited by cyberattacks, thus improving the overall security of the system.
3. Availability issue: using your own data center can directly maintain your system and ensure the availability of it by yourself, whereas if some problems happened in the cloud, you will need to contact the cloud provider to deal with them.
4. Professional training issue: if your system is deployed in the cloud, you will need to train employees with professsional knowledge related to the cloud computing, which can be time-consuming and costly.
5. Cost control issue: Since the financial company usually serve on long-term businesses whose features do not change very often. This means a one-time investment for the data center would be more cost-effective than continuously renting services on the cloud.

Q4

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 Scale

refers to adding multiple computing devices or nodes to the system.

In a data center, administrators traditionally achieved vertical scaling by purchasing a new, more powerful server and discarding or repurposing the old one.

horizontal scale

refers to adding more resources to a single computing device to improve performance.

Scale application horizontally

1. Adding more instances within a server, improve ability to deal with requests simultaneously.
2. Deploy multiple server. Adding more machines in the pool of existing resources, choose the closest and available machines to execute the computation.

Three methods to scale database: Master-slave replication, Clustering, Sharding. For database sharding:

1. Database sharding is an architecture to split the primary database into multiple databases, including two types of sharding techniques, vertical and horizontal sharding.
2. In horizontal sharding, tables are taken out and placed on different machines, each with the same columns but different rows.

Q5 NEW

How could a mobile device benefit from cloud computing? Explain the reasons or provide your arguments supporting the contrary. Discuss several cloud applications for mobile devices; explain which one of the three cloud computing delivery models, SaaS, PaaS, or IaaS, would be used by each one of the applications and why.

1. Integrated data: Mobile cloud computing enables users to quickly and securely collect and integrate data from various sources, regardless of where it resides.
2. shared resources: all data-intensive processes can run from the cloud, which means any mobile device with access to a network can use mobile cloud apps, regardless of the operating system. This means users can enjoy cloud computing with Android or OS devices.
3. Speed and flexible: cloud computing is fast and flexible, which makes it easy for developers to create and share mobile app resources with end-users. Therefore, mobile applications can be built and updated faster.
4. Here are some specific advantages generated by applications designed under mobile cloud computing architecture:

- Applications enjoy better processing power and data storage capacity
- Applications run more efficiently, thus extending battery life
- Applications are more user-friendly and easier to integrate
- Applications are more reliable and scalable

Example Office 365 app, Dropbox application.

SaaS: Office 365 app on mobile devices is SaaS, which provides an online version of MS Office Suite (Office Web Apps) along with SharePoint Server, Exchange Server and Lync Server.

IaaS: Using Dropbox for file storage on mobile devices is an example of an IaaS. You can access, alter and add data as you please, while Dropbox provides the servers to host it. Almost every website you visit is hosted through the cloud, thanks to IaaS models.

Week2

Q6

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.

1. Create a security group
2. Authorise inbound traffic for ssh
3. Create a key pair that will allow you to ssh to the EC2 instance
4. Create the instance and note the instance id
5. Get the public IP address
6. Connect to the instance
7. Look at the instance using the AWS console

Q7

Describe EBS and what features it offers.

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances.

1. EBS storage is allocated in volumes, A volume is a 'virtual disk'
2. A single instance can access multiple volumes.

3. A raw block device, can be attached to an instance.
4. It can be palced in specific availability zones.

Q8 NEW

What is CLI and Boto? What are advantages of using CLI? How does Boto function helps in AWS operation?

CLI stands for command line interface. The AWS CLI is a python program that is written using Boto3. It provides a unified interface to all AWS service APIs.

Boto is a Python interface to Amazon Web Services.

advantages of CLI:

- 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.

Boto functions helps in AWS operation:

Both boto and CLI can access AWS services. However, Boto makes it easier to write Python code that used the AWS Rest API by providing a series of Python Objects that you can manipulate. Compare to boto, marshaling requests from scratch to the AWS CLI will be more time-consuming.

Week3

Q9

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

Virtualisation is the process of using software that simulates hardware functionality to create a virtual system.

Virtualization in a programming language: - It is mostly utilised for application delivery, controlled execution, and portability between different platforms and operating systems. This is a virtual machine that executes a program's byte code which is the result of the compilation process. It generates a binary format that represents the computer code for an abstract architecture. For instance, - Java virtual machine (JVM) and NET provide Common Language Infrastructure (CLI).

Virtualization in the operating system: - It provides the ability to build different and separate execution environments for simultaneously controlled applications. Multiple segregated user space instances are enabled by the OS kernel. For instance - chroot, Jails, OpenVZ, etc.

Virtualization in Hardware: - In hardware virtualization, this enables a program compiled against other hardware to be executed. Guest OS doesn't require complete isolation. The virtualization of address space is a standard aspect of the contemporary OS.

Exam 3

Q10

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

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Docker offers the ability to run multiple applications within the same host OS, sharing underlying resources.

Similarity: Both Docker and VirtualBox are virtualisation tools running on the computer.

Differences: VirtualBox is an virtualization app that create virtual machines that are isolated at the hardware level, whereas Docker is a containerization app that isolates apps at application level.

Q11 NEW

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.

• 1960 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

1. mainframe virtualisation: logically partition mainframe computers into separate virtual machines. These partitions allowed mainframes to run multiple applications and processes at the same time.
2. VMware: Provides full virtualization for hardware.
3. VirtualBox: Provides virtualization for both hardware and software

Week4

Q12

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.

Using AWS CLI

1. 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
```

2. write an entry

```
aws dynamodb put-item \
--table-name Music \
--item '{ "AlbumTitle": {"S": "abc"}, "Artist": {"S": "def"} }' \
--endpoint-url=http://localhost:8000
```

Q13

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.

1. For New objects

- read-after-write consistency
- Synchronous store of data across multiple facilities

2. Updates and deletes could report old data if read before updates complete eventual consistency

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 • Versioning can be enabled for each bucket

Q14

What are the core components of DynamoDB

1. Tables

- Collection of data: People Schemaless

2. Items

- Group of attributes that is uniquely identifiable: Person in People table Physically a JSON document

3. Atrributes

- Fundamental data element
- Limited types of data: Scalar, Document, Sets
- LastName in Person item
- One or more attributes make a primary key (unique)

Q15

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?

1. Select: Region, Versioning, Server access logging, Tags, Object-level logging, Encryption: SSE-S3(AES-256), SSE-KMS
2. Select Properties including

- Users with special permissions of Read and Write
- Grant public read access to the bucket

3. Managing features including

- Life-cycle: Transit objects that are infrequently accessed to cold storage
- Replication: Automatically copy objects to another bucket in a different region
- Analytic: Suggest how to manage objects based on access patterns

Q16 NEW

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

There are a number of indents that happened due to inappropriate permission control of S3 buckets.

- 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
  - Interesting to consider if this is a failure of the users or the system

When this sensitive personal data is compromised, both information security and individual privacy issues may arise. For the companies being affected, it can also result in a crisis of confidentiality, integrity, and availability (CIA).

Week5

Q17

An organisation has 5 departments and has separated out each of the IAM users into separate groups using paths following the pattern companybucket/department1/, *companybucket /department2/*, companybucket /department3/* etc.

Their IAM account names follow the pattern user@department1.company.com, user@department2.company.com etc.

You are tasked with **securing a bucket** that contains a folder for each of 5 departments in an organisation. Only people within a department can write to their own folder. Everyone can read from all folders.

Discuss the principles that you would use to create a policy that would achieve this objective.

Write the policy as a JSON file that you would use.

Note: you can have individual statements for each department.

we first write a statement for all IAM users to have permission to read all resources in the companybucket. Then we write 5 more statements, each of them determine the permission that users in each department can write to their own folder. The policy is as follows:

```
{
  "Version": "2012-10-17",
  "Statement": [
      {
      "Sid": "AllowRead",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3::: companybucket/*",
      "Principal": "*",
      "Condition": {
        "StringLike": {
          "aws:username": "@department1.company.com",
          "aws:username": "@department2.company.com",
          "aws:username": "@department3.company.com",
          "aws:username": "@department4.company.com",
          "aws:username": "@department5.company.com"
        }
      },
      {
      "Sid": "AllowAllInDepartment1",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3::: companybucket/department1/*",
      "Principal": "*",
      "Condition": {
        "StringLike": {
          "aws:username": "@department1.company.com"
        }
      },
      {
      "Sid": "AllowAllInDepartment2",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3::: companybucket/department2/*",
      "Principal": "*",
      "Condition": {
        "StringLike": {
          "aws:username": "@department2.company.com"
        }
      },
      {
      "Sid": "AllowAllInDepartment3",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3::: companybucket/department3/*",
      "Principal": "*",
      "Condition": {
        "StringLike": {
          "aws:username": "@department3.company.com"
        }
      },
      {
      "Sid": "AllowAllInDepartment4",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3::: companybucket/department4/*",
      "Principal": "*",
      "Condition": {
        "StringLike": {
          "aws:username": "@department4.company.com"
        }
      },
      {
      "Sid": "AllowAllInDepartment5",
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": "arn:aws:s3::: companybucket/department5/*",
      "Principal": "*",
      "Condition": {
        "StringLike": {
          "aws:username": "@department5.company.com"
        }
      },



    }
  ]
  }
}
```

Q18

What aspects of security does the OSI Security Architecture X.800 standard cover? Which particular components of this standard does AWS Identity and Access Management deal with?

1. Authentication **MFA** The assurance that the communicating entity is the one that it claims to be

For increased security, configuring multi-factor authentication (MFA) can help protect AWS resources. You can enable MFA for IAM users or the AWS account root user. When you enable MFA for the root user, it affects only the root user credentials. IAM users in the account are distinct identities with their own credentials, and each identity has its own MFA configuration.

1. Access control **ABAC** The prevention of unauthorized use of a resource

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called tags. You can attach tags to IAM resources, including IAM entities (users or roles) and to AWS resources. You can create a single ABAC policy or small set of policies for your IAM principals. These ABAC policies can be designed to allow operations when the principal's tag matches the resource tag. ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

3. Data confidentiality **HSM** The protection of data from unauthorized disclosure.

Four Types

Connection Confidentiality Connectionless Confidentiality Selective-Field Confidentiality Traffic-Flow Confidentiality

4. Data integrity **HSM** The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

Five types

Connection Integrity with Recovery Connection Integrity without Recovery Selective-Field Connection Integrity Connectionless Integrity Selective-Field Connectionless Integrity 5. Nonrepudiation **HSM** Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication

Two types

Nonrepudiation, Origin Nonrepudiation, Destination

6. Availability – resource accessible/usable

HSM AWS CloudHSM provides hardware security modules in the AWS Cloud. A hardware security module (HSM) is a computing device that processes cryptographic operations and provides secure storage for cryptographic keys.

Generate, store, import, export, and manage cryptographic keys, including symmetric keys and asymmetric key pairs.

Use symmetric and asymmetric algorithms to encrypt and decrypt data.

Use cryptographic hash functions to compute message digests and hash-based message authentication codes (HMACs).

Cryptographically sign data (including code signing) and verify signatures.

Generate cryptographically secure random data.

Q19

Name 3 of the keys that you would find in a Policy. Explain their role. An example of a key is "Version" that specifies the version of the policy syntax and is normally "Version": "2012-10-17"

Sid (Optional) The statement identifier (Sid) an arbitrary string you can use to describe the statement. The Sid in a key policy can include spaces. (You can't include spaces in an IAM policy Sid element.)

Effect (Required) Determines whether to allow or deny the permissions in the policy statement. Valid values are Allow or Deny. If you don't explicitly allow access to a KMS key, access is implicitly denied. You can also explicitly deny access to a KMS key. You might do this to make sure that a user cannot access it, even when a different policy allows access.

Action (Required) Specify the API operations to allow or deny. For example, the kms:Encrypt action corresponds to the AWS KMS Encrypt operation. You can list more than one action in a policy statement. For more information, see Permissions reference.

Resource (Required) In a key policy, the value of the Resource element is "*", which means "this KMS key." The asterisk (*"*") identifies the KMS key to which the key policy is attached.