Program : Computer Science Engineering **Course : CLOUD COMPUTING** **Course Code : 20CS53I**	**Scheme of Valuation Jan/Feb-2023**	**Semester: Vth** **Duration :3 Hrs** **Max Marks : 100**

Instruction to the Candidate: Answer one full question from each section.

Q. No	**Question**	
Section-I		
1.a	Explain Cloud Deployment models.	Explanation of each model 4x2.5
b	Explain the following. i. AWS Regions and Availability zones. ii. List IAM policy types supported by AWS.	explanation 2.5+2.5 any five policy 1 mark each
2.a	List and explain EC2 instance types.	List-1, explain 6 types 6x1.5
b	Write steps involved in creating an AWS EC2 Instance.	Write all steps=10
Section-II		
3.a	Write a note on: i. VPC ii. Internet gateway iii. Bastion host iv. NAT gateway v. subnet	Explanation 5x2
b	Write steps to create a VPC peering connection with VPCs in the same account and Region.	Write all steps=10
4.a	List Elastic Load Balancers types and explain Application Load Balancer.	List-4, Explanation-6
b	List and explain Amazon RDS Database Engines.	List-1, explain 6 DB engines 6x1.5
Section-III		
5.a	Write steps to configure an Amazon S3 bucket for static website hosting.	Write all steps=10
b	Explain different Amazon S3 object-level storage classes.	List-1,Explanation 6x1.5
6.a	Identify and explain the AWS Terminologies: i. Bucket ii. Key iii. Versioning iv. Object v. Bucket policy	Explanation 5x2
b	Explain Content Delivery Network and list its advantages.	Explaination- 5, advantages-5

Section-IV		
7.a	Explain any five Kubernetes components.	Explaining any five 5x2=10
b	With a neat diagram explain the working of Elastic Beanstalk.	Diagram-4, Explanation-6
8.a	Compare AWS Cloud Trail Vs Cloud Watch.	Any 5 differences- 5x2=10
b	Explain the following: i. CI/CD in AWS ii. Defence in depth in security	Explanation-5 Explanation-5
Section-V		
9.a	List and brief the purpose of Azure storage data services.	List-5, Explanation -5
b	Write a note on Virtual Machine Security Best Practices in Azure Cloud.	Any 10 Security Best Practices 10x1
10.a	Explain different types of Azure storage accounts.	Explanation 4x2.5
b	Write the steps for creating Azure functions.	Write all steps-10

VI Semester Diploma Examination, January/February-2023

CLOUD COMPUTING- 20CS53I

MODEL ANSWERS

1.a) Explain Cloud Deployment models. 10

Deployment Models

The cloud deployment model identifies the specific type of cloud environment based on ownership, scale, and access, as well as the cloud's nature and purpose. The location of the servers you're utilizing and who controls them are defined by a cloud deployment model.

Different types of cloud computing deployment models are:

1. Public cloud
2. Private cloud
3. Hybrid cloud
4. Community cloud

Public cloud :

- ❖ It is accessible to the public. Public deployment models in the cloud are perfect for organizations with growing and fluctuating demands.
- ❖ It also makes a great choice for companies with low-security concerns. Thus, you pay a cloud service provider for networking services, compute virtualization & storage available on the public internet.
- ❖ It is also a great delivery model for the teams with development and testing.
- ❖ Its configuration and deployment are quick and easy, making it an ideal choice for test environments.

Advantages of Public Cloud Model:

- ✓ **Minimal Investment: No setup cost:** The entire infrastructure is fully subsidized by the cloud service providers, thus there is no need to set up any hardware.
- ✓ **Infrastructure Management is not required:** Using the public cloud does not necessitate infrastructure management.
- ✓ **No maintenance:** The maintenance work is done by the service provider (Not users).
- ✓ **Dynamic Scalability:** To fulfill your company's needs, on-demand resources are accessible.

Disadvantages of Public Cloud Model:

- ✓ **Less secure:** Public cloud is less secure as resources are public so there is no guarantee of high-level security.
- ✓ **Low customization:** It is accessed by many public so it can't be customized according to personal requirements.

Private Cloud:

- ❖ The private cloud deployment model is the exact opposite of the public cloud deployment model. It's a one-on-one environment for a single user (customer).
- ❖ There is no need to share your hardware with anyone else. The distinction between private and public clouds is in how you handle all of the hardware.
- ❖ It is also called the "internal cloud" & it refers to the ability to access systems and services within a given border or organization.
- ❖ The cloud platform is implemented in a cloud-based secure environment that is protected by powerful firewalls and under the supervision of an organization's IT department.
- ❖ The private cloud gives greater flexibility of control over cloud resources.

Advantages of Private Cloud Model:

- ✓ **Better Control:** You are the sole owner of the property. You gain complete command over service integration, IT operations, policies, and user behavior.
- ✓ **Data Security and Privacy:** It's suitable for storing corporate information to which only authorized staff have access. By segmenting resources within the same infrastructure, improved access and security can be achieved.
- ✓ **Supports Legacy Systems:** This approach is designed to work with legacy systems that are unable to access the public cloud.
- ✓ **Customization:** Unlike a public cloud deployment, a private cloud allows a company to tailor its solution to meet its specific needs.

Disadvantages of Private Cloud Model:

- ✓ **Less scalable:** Private clouds are scaled within a certain range as there is less number of clients.
- ✓ **Costly:** Private clouds are more costly as they provide personalized facilities.

Hybrid Cloud:

- ❖ By bridging the public and private worlds with a layer of proprietary software, hybrid cloud computing gives the best of both worlds.
- ❖ With a hybrid solution, you may host the app in a safe environment while taking advantage of the public cloud's cost savings.
- ❖ Organizations can move data and applications between different clouds using a combination of two or more cloud deployment methods, depending on their needs.

Advantages of Hybrid Cloud Model:

- ✓ **Flexibility and control:** Businesses with more flexibility can design personalized solutions that meet their particular needs.
- ✓ **Cost:** Because public clouds provide scalability, you'll only be responsible for paying for the extra capacity if you require it.
- ✓ **Security:** Because data is properly separated, the chances of data theft by attackers are considerably reduced.

Disadvantages of Hybrid Cloud Model:

- ✓ **Difficult to manage:** Hybrid clouds are difficult to manage as it is a combination of both public and private cloud. So, it is complex.
- ✓ **Slow data transmission:** Data transmission in the hybrid cloud takes place through the public cloud so latency occurs.

Community Cloud

- ❖ It allows systems and services to be accessible by a group of organizations.
- ❖ It is a distributed system that is created by integrating the services of different clouds to address the specific needs of a community, industry, or business.
- ❖ The infrastructure of the community could be shared between the organization which has shared concerns or tasks.
- ❖ It is generally managed by a third party or by the combination of one or more organizations in the community.

Advantages of Community Cloud Model:

- ✓ **Cost Effective:** It is cost-effective because the cloud is shared by multiple organizations or communities.
- ✓ **Security:** Community cloud provides better security.
- ✓ **Shared resources:** It allows you to share resources, infrastructure, etc. with multiple organizations.
- ✓ **Collaboration and data sharing:** It is suitable for both collaboration and data sharing.

Disadvantages of Community Cloud Model:

- ✓ **Limited Scalability:** Community cloud is relatively less scalable as many organizations share the same resources according to their collaborative interests.
- ✓ **Rigid in customization:** As the data and resources are shared among different organizations according to their mutual interests if an organization wants some changes according to their needs they cannot do so because it will have an impact on other organizations.

1.b) Explain the following.

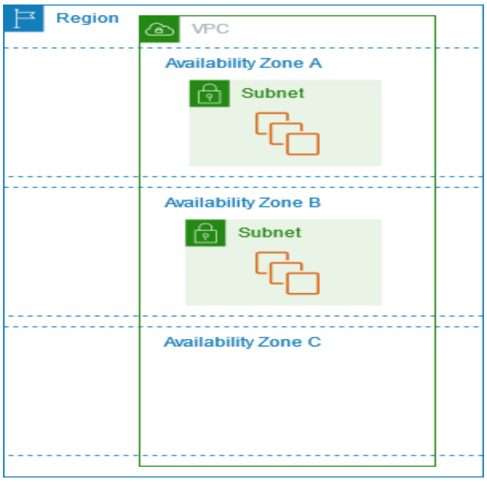
i. AWS Regions and Availability zones. 5

AWS Regions

- ❖ Each AWS Region is designed to be isolated from the other AWS Regions. This design achieves the greatest possible fault tolerance and stability.
- ❖ When you view your resources, you see only the resources that are tied to the AWS Region that you specified.
- ❖ This is because AWS Regions are isolated from each other, and we don't automatically replicate resources across AWS Regions.
- ❖ Each AWS Region consists of a minimum of three, isolated, and physically separate AZs within a geographic area. Unlike other cloud providers, who often define a region as a single data center, the multiple AZ design of every AWS Region offers advantages for customers.

Availability zones:

- ❖ Each Region has multiple, isolated locations known as *Availability Zones*. The code for Availability Zone is its Region code followed by a letter identifier. For example, us-east-1a.
- ❖ When you launch an instance, you select a Region and a virtual private cloud (VPC), and then you can either select a subnet from one of the Availability Zones or let us choose one for you.
- ❖ If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests.
- ❖ You can also use Elastic IP addresses to mask the failure of an instance in one Availability Zone by rapidly remapping the address to an instance in another Availability Zone.
    - ✓ The following diagram illustrates multiple Availability Zones in an AWS Region. Availability Zone A and Availability Zone B each have one subnet, and each subnet has instances.
    - ✓ Availability Zone C has no subnets, therefore you can't launch instances into this Availability Zone.



ii. List IAM policy types supported by AWS. 5

AWS supports six types of policies:

1. Identity-based policies
2. Resource-based policies
3. Permissions boundaries
4. Organizations SCPs
5. Access Control Lists
6. Session policies.

2. a) List and explain EC2 instance types. 10

EC2 Instances types:

1. General Purpose Instances

- ❖ General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads.
- ❖ General purpose instances are optimized to have a high number of CPU cores, on-demand storage and memory.
- ❖ These instances are ideal for applications that use these resources in equal proportions such as web servers and software development and testing.

2. Compute Optimized Instances

- ❖ Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors.
- ❖ Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.

3. Memory-Optimized Instances

- ❖ Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.
- ❖ Examples include high performance databases and distributed cache, in-memory analytics.

4. Accelerated Computing Instances

- ❖ Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

5. Storage Optimized Instances

- ❖ Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage.
- ❖ They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

6. HPC Optimized

- ❖ High performance computing (HPC) instances are built to offer the best price performance for running HPC workloads at scale on AWS.
- ❖ HPC instances are ideal for applications that benefit from high-performance processors such as large, complex simulations and deep learning workloads.

2.b)Write steps involved in creating an AWS EC2 Instance. 10

Steps to Create an AWS EC2 instance

1. Go to the AWS Management Console (https://aws.amazon.com/console/). On services, tab search for EC2 and click on launch instances for creating an EC2 Instance.
2. First, enter a name tag for the instance.
3. Next, you need to choose a base image for your EC2 instance i.e. operating system of your instance.
4. Next, we need to choose an instance type. Select T2 micro which is free tier eligible.
5. Next, create a key pair to log into your instance. So this is necessary if we use the SSH utility to access our instance. We could proceed without a key pair, but for now, let’s go ahead and create a new key pair.
   a. Give a name to Key Pair.
   b. Then you need to choose a key pair type (RSA encrypted or ED25519 encrypted), here we’ll be using the RSA encrypted.
   c. And then we have to select key pair formats. If you have Mac or Linux or Windows 10 then you can use the .pem format.
   d. When you create a key pair it will be downloaded automatically to your pc.
6. Next, we have to go into network settings. Leave the network and subnet settings to default. Select auto assign public IP enabled and then create security group which is to be attached to our instance.
7. Next, we need to configure the storage for this instance let’s have an 8gb gp2 root volume.
8. Finally, we can review everything we have created and launch this instance.

SECTION-II

3. a) Write a note on: 10

i. VPC
ii. Internet gateway
iii. Bastion host
iv. NAT gateway
v. subnet

i. VPC :

- A VPC is a virtual network that closely resembles a traditional network that you'd operate in your own data center.
- Amazon Virtual Private Cloud (Amazon VPC) is a service that lets you provision a logically isolated section of the AWS Cloud (called a virtual private cloud, or VPC) where you can launch your AWS resources.
- Amazon VPC gives you control over your virtual networking resources, including the selection of your own IP address range, the creation of subnets, and the configuration of route tables and network gateways.

ii. Internet gateway:

- An internet gateway is redundant, horizontally scaled, and is a highly available VPC component.
- It enables communication between instances in your VPC and the internet.
- Therefore, it imposes no availability risks or bandwidth constraints on your network traffic. To give your VPC the ability to connect to the internet, you need to attach an internet gateway.
- Only one internet gateway can be attached per VPC.
- Attaching an internet gateway is the first stage in permitting internet access to instances in your VPC.

iii. Bastion host:

- A **bastion host** is a server whose purpose is to provide access to a private network from an external network, such as the Internet.
- Bastion hosts are instances that sit within your public subnet and are typically accessed using SSH or RDP.
- Once remote connectivity has been established with the bastion host, it then acts as a 'jump' server, allowing you to use SSH or RDP to log in to other instances (within private subnets) deeper within your VPC.
- When properly configured through the use of security groups and Network ACLs (NACLs), the bastion essentially acts as a bridge to your private instances via the internet.

iv. NAT gateway:

- A NAT gateway is a Network Address Translation (NAT) service. You can use a NAT gateway so that instances in a private subnet can connect to services outside your VPC but external services cannot initiate a connection with those instances.
- Once created, you need to update the route table associated with your private subnet to point internet-bound traffic to the NAT gateway. This way, the instances in your private subnet can communicate with the internet.

v. subnet:

- AWS defines a subnet as a range of IP addresses in your VPC. You can launch AWS resources into a selected subnet.
- A public subnet can be used for resources connected to the internet and a private subnet for resources not connected to the internet.
- There are two types of subnets: public and private.

3.b) Write steps to create a VPC peering connection with VPCs in the same account and Region. **10**

Step to create a VPC peering connection with VPCs in the same account and Region

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Peering connections**.
3. Choose **Create peering connection**.
4. Configure the following information, and choose **Create Peering Connection** when you are done:
   - **Peering connection name tag**: You can optionally name your VPC peering connection.
   - **VPC (Requester)**: Select the VPC in your account with which you want to create the VPC peering connection.
   - Under **Select another VPC to peer with**: Ensure **My account** is selected, and select another of your VPCs.
   - (Optional) To add a tag, choose **Add new tag** and enter the tag key and value.
5. In the confirmation dialog box, choose **OK**.
6. Select the VPC peering connection that you've created, and choose **Actions**, **Accept Request**.
7. In the confirmation dialog, choose **Yes, Accept**. A second confirmation dialog displays; choose **Modify my route tables now** to go directly to the route tables page or choose **Close** to do this later.

4.a) List Elastic Load Balancers types and explain Application Load Balancer. 10

Elastic Load Balancer provides four types of load balancers:

1. Application Load Balancers
2. Network Load Balancers,
3. Gateway Load Balancers
4. Classic Load Balancers.

Application Load Balancer:-

- ✓ Application Load Balancer operates at the request level (layer 7), routing traffic to targets (EC2 instances, containers, IP addresses, and Lambda functions) based on the content of the request.
- ✓ Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including micro services and container-based applications.
- ✓ Application Load Balancer simplifies and improves the security of your application, by ensuring that the latest SSL/TLS ciphers and protocols are used at all times.

Application Load Balancer has the following benefits:

- ➢ Support for Path conditions
- ➢ Support for Host conditions.
- ➢ Support for routing based on fields in the request
- ➢ Support for routing requests to multiple applications on a single EC2 instance.
- ➢ Support for returning a custom HTTP response.

4 .b) List and explain Amazon RDS Database Engines. 10

Amazon RDS currently supports the following engines:

- ❖ MariaDB
- ❖ Microsoft SQL Server
- ❖ MySQL
- ❖ Oracle
- ❖ PostgreSQL
- ❖ Amazon Aurora

Let us explore each of these databases engines in depth.

PostgreSQL:

- ❖ Being an open-source relational database, it’s the most preferred database engine for developers. Amazon RDS for PostgreSQL provides the same functionalities as the traditional PostgreSQL database. There is no surprise that the codes, applications, and tools are readily used with the current databases and can also be integrated with Amazon RDS for seamless communication. In addition, users can also use Postgre SQL to make databases scalable and easy to deploy in a cost-efficient manner. With the help of AWS RDS, the hardware can be made flexible and resizable in terms of capacity.

MySQL:

- ❖ MySQL is the world’s most-liked open-source relational database and Amazon RDS provides simple and easy ways to set up, operate, and scale MySQL deployments in the AWS cloud. Users can use the same code written on local MYSQL instances because RDS for MYSQL covers all versions of MYSQL.

MariaDB:

- Also, MariaDB is an open-source relational database brought into existence by the original developers of MYSQL. Amazon RDS supports all versions of MariaDB server that makes it easier to use existing code, applications and tools with Amazon RDS.

Oracle:

- Amazon RDS specifically designed for Oracle is a commercial database that is managed on our own which supports licensing models and Bring-Your- Own-License (BYOL)". Since the Oracle database Software is licensed by WAS so, it is not required for users to spend extra on Oracle licensing or purchasing. Amazon RDS enables developers to provide extra attention to creativity and app development to manage database administration tasks on their own.

Microsoft SQL Server:

- Microsoft powered SQL server-based relational database management system which supports multiple versions of SQL server from (2012, 2014, 2016, 2017 and 2019) which is also inclusive of Express, Web, Standard and Enterprise form of Amazon RDS. The code can be deployed using AWS RDS easily within minutes in a cost-friendly manner with a compact computational capacity.
- This is a licensing model supported by Amazon RDS for SQL Server and hence it is not necessary to purchase any Microsoft SQL Server licenses additionally. Also, Amazon RDS for SQL facilitates users with two options Standard storage or provisioned IOPS for rapid, predictable, and consistent input-output and performance

Amazon Aurora:

- Amazon Aurora is a relational database management system (RDBMS) built for the cloud with full MySQL and PostgreSQL compatibility.
- Aurora gives you the performance and availability of commercial-grade databases at one-tenth the cost.
- The code, tools, and applications you use today with your existing MySQL and PostgreSQL databases can be used with Aurora.
- Aurora includes a high-performance storage subsystem. The underlying storage grows automatically as needed. An Aurora cluster volume can grow to a maximum size of 128 tebibytes (TiB).
- Aurora also automates and standardizes database clustering and replication, which are typically among the most challenging aspects of database configuration and administration.

SECTION-III

5.a) Write steps to configure an Amazon S3 bucket for static website hosting. 10

steps to configure an Amazon S3 bucket for static website hosting.

Step 1: Create a S3 Bucket.

a. Choose Create bucket
b. Enter Bucket name and choose region. An S3 bucket name is globally unique.
c. Public access to buckets is blocked by default.
d. In the Object Ownership section, select ACLs enabled, then verify Bucket owner preferred is selected.
e. Clear Block all public access, then select the box that states I acknowledge that the current settings may result in this bucket and the objects within becoming public.
f. Choose Create bucket.

Step 2: Upload objects to your S3 Bucket.

a. Open the bucket created and upload objects.
b. Choose Upload
c. Choose Add files
d. Locate and select all website file that you have developed.
e. Choose Upload Choose Close.

Step 3: Enabling access to the objects

a. Objects that are stored in Amazon S3 are private by default. You need to public all the objects uploaded to bucket. Select all objects.
b. In the Actions menu, choose Make public via ACL.
c. Choose Make public

Step 4: Enable Static Website Hosting

a. Move to the Properties Tab. Scroll to the Static website hosting panel.
b. Choose Edit
c. Configure the following settings:
   - Static web hosting: Enable
   - Hosting type: Host a static website
   - Index document: index.html
   - Error document: error.html
d. Choose Save changes
e. In the Static website hosting panel, choose the link under Bucket website endpoint.
f. Open browser and paste it in address bar. Tour website has been hosted and now it is accessible.

5.b) Explain different Amazon S3 object-level storage classes. 10

Amazon S3 offers following storage classes that are designed for different use cases.

1. Amazon S3 Standard –

Amazon S3 Standard is designed for frequently accessed data.Because it delivers low latency and high throughput.

2. Amazon S3 Intelligent – Tiering –

The Amazon S3 Intelligent - Tiering storage class is designed to optimize costs by automatically moving data to the most cost - effective access tier, moves the objects that have not been accessed for 30 consecutive days to the infrequent access tier. If an object in the infrequent access tier is accessed, it is automatically moved back to the frequent access tier. No additional fees when objects are moved between access tiers.

3. Amazon S3 Standard - Infrequent Access (Amazon S3 Standard-IA) –

The Amazon S3 Standard-IA storage class is used for data that is accessed less frequently, but requires rapid access when needed.

4. Amazon S3 One Zone - Infrequent Access (Amazon S3 One Zone - IA) –

Amazon S3 One Zone- IA is for data that is accessed less frequently, but requires rapid access when needed. Unlike other Amazon S3 storage classes, which store data in a minimum of three Availability Zones, Amazon S3 One Zone - IA stores data in a single Availability Zone and it costs less than Amazon S3 Standard - IA.

5. Amazon S3 Glacier –

Amazon S3 Glacier is a secure, durable, and low-cost storage class for data archiving.

6. Amazon S3 Glacier Deep Archive –

Amazon S3 Glacier Deep Archive is the lowest – cost storage class for Amazon S3. It supports long - term retention and digital preservation for data that might be accessed once or twice in a year.

6.a) Identify and explain the AWS Terminologies: 10

i. Bucket
ii. Key
iii. Versioning
iv. Object
v. Bucket policy

i. Bucket

- A *bucket* is a container for objects. An Amazon S3 bucket is a public cloud storage resource available in Amazon Web Services(AWS) Simple Storage Service (S3), an object storage offering.
- Amazon S3 buckets, which are similar to file folders, store objects, which consist of data and its descriptive metadata.
- Once the bucket has been created, the user then selects a tier for the data, with different S3 tiers having different levels of redundancy, prices and accessibility. One bucket can store objects from different S3 storage tiers.

ii. Key

- The *object key* (or key name) uniquely identifies the object in an Amazon S3 bucket.
- *Object metadata* is a set of name-value pairs. For more information about object metadata, see Working with object metadata.
- The object key name is a sequence of Unicode characters with UTF-8 encoding of up to 1,024 bytes long.
- When you create an object, you specify the key name, which uniquely identifies the object in the bucket.

iii. Versioning

- Versioning in Amazon S3 is a means of keeping multiple variants of an object in the same bucket.
- You can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets.
- With versioning you can recover more easily from both unintended user actions and application failures. After versioning is enabled for a bucket, if Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of those objects.

iv. Object

- An *object* is a file and any metadata that describes that file.
- To store an object in Amazon S3, you create a bucket and then upload the object to the bucket.
- When the object is in the bucket, you can open it, download it, and move it.
- When you no longer need an object or a bucket, you can clean up your resources.

v. Bucket policy

- A bucket policy is a resource-based policy that you can use to grant access permissions to your Amazon S3 bucket and the objects in it.
- Only the bucket owner can associate a policy with a bucket. The permissions attached to the bucket apply to all of the objects in the bucket that are owned by the bucket owner.
- These permissions do not apply to objects that are owned by other AWS accounts.

- ❖ Bucket policies use JSON-based IAM policy language.
- ❖ You can use bucket policies to add or deny permissions for the objects in a bucket.
- ❖ Bucket policies can allow or deny requests based on the elements in the policy.

6.b) Explain Content Delivery Network and list its advantages. 10

- ❖ A content delivery network (CDN) is a network of interconnected servers that speeds up webpage loading for data-heavy applications.
- ❖ When a user visits a website, data from that website's server has to travel across the internet to reach the user's computer.
- ❖ If the user is located far from that server, it will take a long time to load a large file, such as a video or website image. Instead, the website content is stored on CDN servers geographically closer to the users and reaches their computers much faster.

Why is a CDN important?

- ❖ The primary purpose of a content delivery network (CDN) is to reduce latency, or reduce the delay in communication created by a network's design.
- ❖ Because of the global and complex nature of the internet, communication traffic between websites (servers) and their users (clients) has to move over large physical distances.
- ❖ A CDN improves efficiency by introducing intermediary servers between the client and the website server. These CDN servers manage some of the client-server communications.
- ❖ They decrease web traffic to the web server, reduce bandwidth consumption, and improve the user experience of your applications.

- **Advantages of CDN.**
- ✓ **Faster delivery of content**
- ✓ **More simultaneous users**
- ✓ **Constant availability**
- ✓ **Reliable delivery of content**
- ✓ **Control over asset delivery**
- ✓ **Protection against traffic spikes**

SECTION-IV

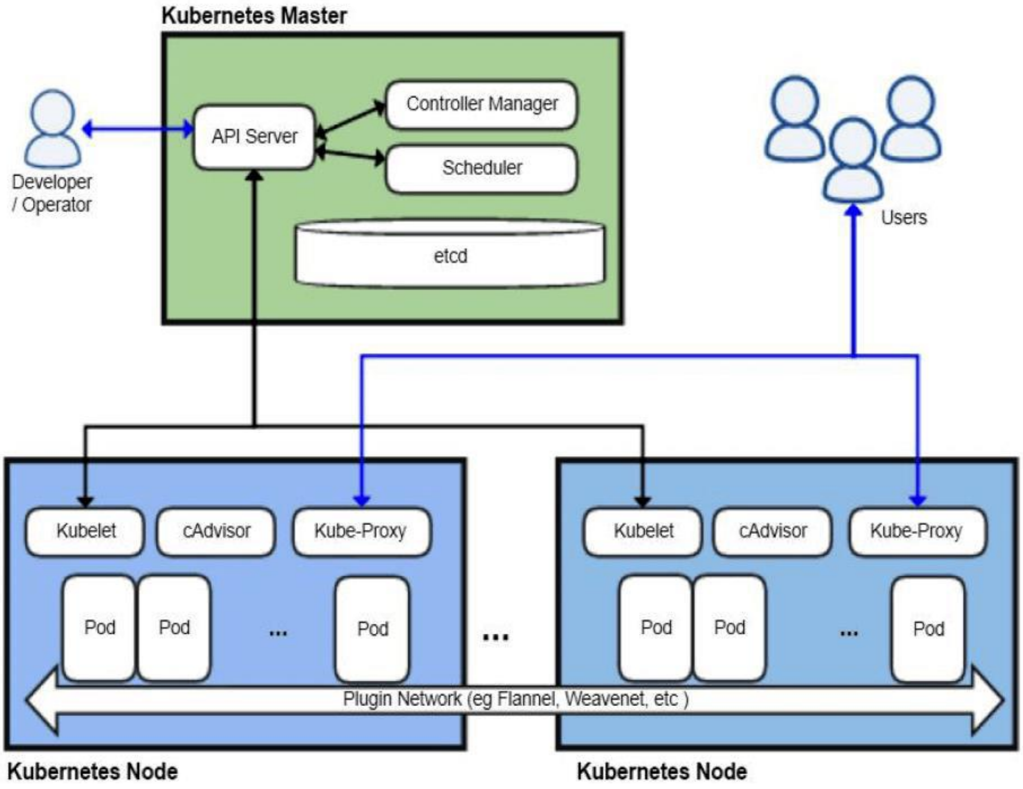
7.a) Explain any five Kubernetes components. 10

Kubernetes Components:

- ❖ The master server consists of various components including a kube-apiserver, an etcd storage, a kube-controller-manager, a cloud-controller-manager, a kube-scheduler, and a DNS server for Kubernetes services. Node components include kubelet and kube-proxy on top of Docker.

Master Node Components:

- ❖ **Kube API-server** performs all the administrative tasks on the master node. A user sends the rest commands as YAML/JSON format to the API server, then it processes and executes them. The Kube API-server is the front end of the Kubernetes control plane.
- ❖ **etcd** is a distributed key-value store that is used to store the cluster state. Kubernetes stores the file in a database called the **etcd**. Besides storing the cluster state, etcd is also used to store the configuration details such as the subnets and the config maps.



- **Read More:** Kubernetes for Testers.
- **Kube-scheduler** is used to schedule the work to different worker nodes. It also manages the new requests coming from the API Server and assigns them to healthy nodes.
- **Kube Controller Manager** task is to obtain the desired state from the API Server. If the desired state does not meet the current state of the object, then the corrective steps are taken by the control loop to bring the current state the same as the desired state.

Node (worker) components:

Below are the main components found on a (worker) node:

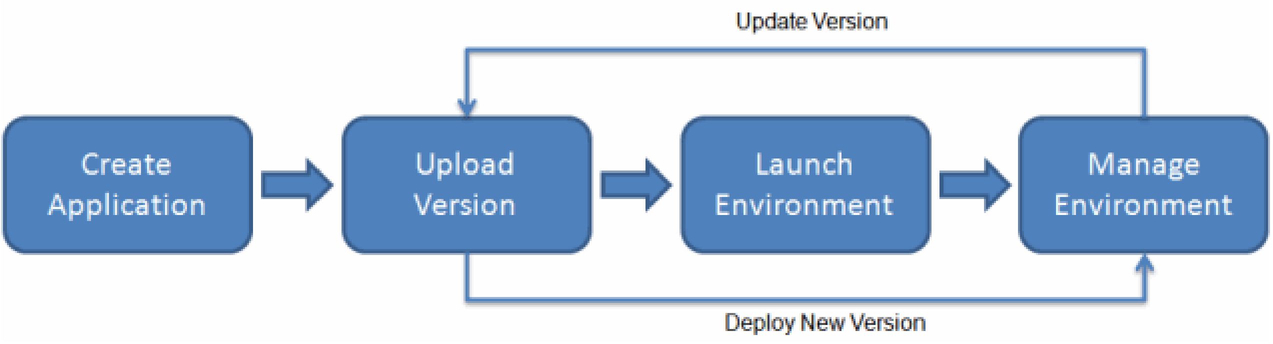
- **kubelet** – the main service on a node, regularly taking in new or modified pod specifications (primarily through the kube-apiserver) and ensuring that pods and their containers are healthy and running in the desired state. This component also reports to the master on the health of the host where it is running.
- **kube-proxy** – a proxy service that runs on each worker node to deal with individual host subnetting and expose services to the external world. It performs request forwarding to the correct pods/containers across the various isolated networks in a cluster.

7.**b) With a neat diagram explain the working of Elastic Beanstalk. 10**

- With Elastic Beanstalk, you can quickly deploy and manage applications in the AWS Cloud without having to learn about the infrastructure that runs those applications.
- Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring.

Working of Elastic Beanstalk.

- ✓ To use Elastic Beanstalk, you create an application, upload an application version in the form of an application source bundle (for example, a Java .war file) to Elastic Beanstalk, and then provide some information about the application.
- ✓ Elastic Beanstalk automatically launches an environment and creates and configures the AWS resources needed to run your code.
- ✓ After your environment is launched, you can then manage your environment and deploy new application versions. The following diagram illustrates the workflow of Elastic Beanstalk.



- ✓ After you create and deploy your application, information about the application—including metrics, events, and environment status—is available through the Elastic Beanstalk console, APIs, or Command Line Interfaces, including the unified AWS CLI.

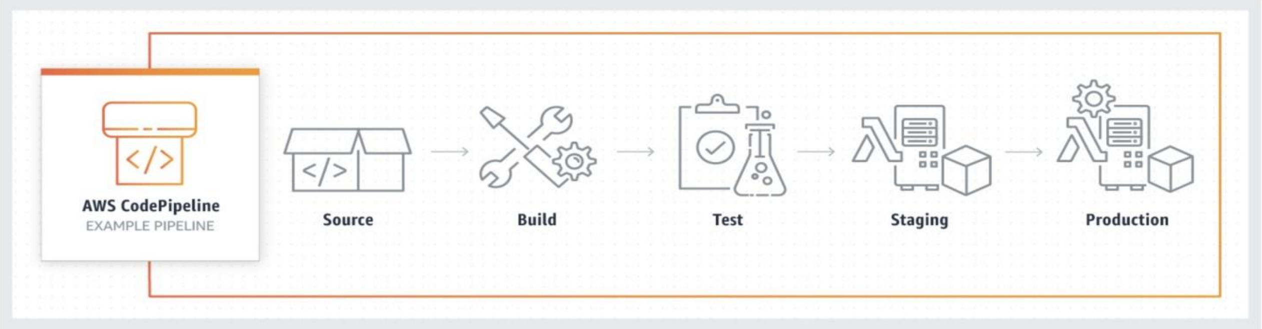
8.a) Compare Cloud Trail Vs Cloud Watch **10**

Cloud Watch	Cloud Trail
CloudWatch is a monitoring service for AWS resources and applications	CloudTrail is a web service that records API activity in your AWS account
CloudWatch offers free basic monitoring for your resources by default, such as EC2 instances, EBS volumes, and RDS DB instances	CloudTrail is also enabled by default when you create your AWS account.
With CloudWatch, you can collect and track metrics, collect and monitor log files, and set alarms	CloudTrail logs information on who made a request, the services used, the actions performed, parameters for the actions, and the response elements returned by the AWS service
CloudWatch delivers metric data in 5 minutes periods for basic monitoring and 1 minute periods for detailed monitoring	CloudTrail delivers an event within 15 minutes of the API call
CloudWatch Logs reports on application logs	CloudTrail Logs provide you specific information on what occurred in your AWS account
CloudWatch Events is a near real time stream of system events describing changes to your AWS resources	CloudTrail focuses more on AWS API calls made in your AWS account

8.b) Explain the following:

i. CI/CD in AWS 05

- ***Continuous Integration*** **(CI)** is a software process in which developers regularly push their code into a central repository such as AWS CodeCommit or GitHub
- ***Continuous Delivery*** **(CD)** is a software process in which artifacts are deployed to the test environment, staging environment, and production environment.
- CI/CD can be pictured as a pipeline, where new code is submitted on one end, tested over a series of stages (source, build, test, staging, and production), and then published as production-ready code.



CICD pipeline overview

- ✓ Each stage of the CI/CD pipeline is structured as a logical unit in the delivery process. Each stage acts as a gate that vets a certain aspect of the code.
- ✓ As the code progresses through the pipeline, the assumption is that the quality of the code is higher in the later stages, because more aspects of it continue to be verified.
- ✓ Problems uncovered in an early stage stop the code from progressing through the pipeline.
- ✓ Results from the tests are immediately sent to the team, and all further builds and releases are stopped if software does not pass the stage.
- ✓ AWS brings in a complete set of CI/CD developer tools to accelerate software development and release cycles.
- ✓ AWS Code Pipeline automates the build, test, and deploy phases of the release process every time there is a code change, based on the defined release model.

ii. Defence in depth in security **05**

- Defense-in-depth represents the use of multiple security defenses to help mitigate the risk of security threats, if one component of the defense is being compromised.
- An example, could be an antivirus software installed on individual VM when there is already a virus protection on the firewalls within the same environment.
- Different security products from multiple vendors may be deployed to defend different potential vulnerable resources within the network.
- Defense-in-depth is an information assurance strategy in which multiple layers of defense are placed throughout the system.
- For this reason, it is also known as a “layered approach to security”. Because there are multiple measures of security at different levels, defense-in-depth gives additional time to detect and respond to an attack. This reduces the scope of a security breach.
- However, the overall cost of deploying defense-in-depth is often higher, compared to single-layered security mechanisms

SECTION-V

9.a) List and brief the purpose of Azure storage data services. 10

List of Azure storage data services.

- Azure Blobs
- Azure Files
- Azure Queues
- Azure Tables
- Azure Disks

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud.

The Azure Storage platform includes the following data services:

- Azure Blobs: A massively scalable object store for text and binary data. Also includes support for big data analytics through Data Lake Storage Gen2. **Azure Blob** storage is a service that stores unstructured data in the cloud as objects/**blobs**. **Blob** storage can store any type of text or binary data, such as a document, media file, or application installer. **Blob** storage is also referred to as object storage.
- Azure Files: Managed file shares for cloud or on-premises deployments. A **File Storage** share is an SMB **file** share in **Azure**. All directories and **files** must be created in a parent share. An account can contain an unlimited number of shares, and a share can store an unlimited number of **files**, up to the 5 TB total capacity of the **file** share.
- Azure Queues: A messaging store for reliable messaging between application components. **Azure Queue** storage is a service for storing large numbers of messages that can be accessed from anywhere in the world via authenticated calls using HTTP or HTTPS. A single **queue** message can be up to 64 KB in size, and a **queue** can contain millions of messages, up to the total capacity limit of a storage account.
- Azure Tables: A NoSQL store for schemaless storage of structured data. The **Azure Table** storage service stores large amounts of structured data. The service is a NoSQL datastore which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data.
- Azure Disks: Block-level storage volumes for Azure VMs. You can think of it like a physical disk in an on-premises server but, virtualized. Azure-managed disks are stored as page blobs, which are a random IO storage object in Azure. We call a managed disk 'managed' because it is an abstraction over page blobs, blob containers, and Azure storage accounts.

9.b) Write a note on Virtual Machine Security Best Practices in Azure Cloud. 10

VM Security Best Practices

- ✓ Use Azure Secure Score in Azure Security Center as your guide.
- ✓ Isolate management ports on virtual machines from the Internet and open them only when required
- ✓ Use complexity for passwords and user account names
- ✓ Keep the operating system patched

- ✓ Keep third-party applications current and patched
- ✓ Actively monitor for threats
- ✓ Azure Backup Service.
- ✓ Protect VMs by using authentication and access control.
- ✓ Use multiple VMs for better availability.
- ✓ Protect against malware.
- ✓ Manage your VM updates.
- ✓ Manage your VM security posture.
- ✓ Monitor VM performance.
- ✓ Encrypt your virtual hard disk files.
- ✓ Restrict direct internet connectivity.

10.a) Explain different types of Azure storage accounts. 10

Types of Storage Accounts:

Azure Storage offers several types of storage accounts. Each type supports different features and has its own pricing model. The following table describes the types of storage accounts recommended by Microsoft for most scenarios. All of these use the Azure Resource Manager deployment model.

Type of storage account	**Supported storage services**	**Redundancy options**	**Usage**
Standard general-purpose v2	Blob Storage (including Data Lake Storage[1]), Queue Storage, Table Storage, and Azure Files	Locally redundant storage (LRS) / geo-redundant storage (GRS) / read-access geo-redundant storage (RA-GRS) Zone-redundant storage (ZRS) / geo-zone-redundant storage (GZRS) / read-access geo-zone-redundant storage (RA-GZRS)2	Standard storage account type for blobs, file shares, queues, and tables. Recommended for most scenarios using Azure Storage. If you want support for network file system (NFS) in Azure Files, use the premium file shares account type.
Premium block blobs3	Blob Storage (including Data Lake Storage1)	LRS ZRS2	Premium storage account type for block blobs and append blobs. Recommended for scenarios with high transaction rates or that use smaller objects or require consistently low storage latency.

Premium file shares3	Azure Files	LRS ZRS2	Premium storage account type for file shares only. Recommended for enterprise or high-performance scale applications. Use this account type if you want a storage account that supports both Server Message Block (SMB) and NFS file shares.
Premium page blobs[3]	Page blobs only	LRS	Premium storage account type for page blobs only. Learn more about page blobs and sample use cases.

10.b) Write steps for creating Azure functions 10

You must have a function app to host the execution of your functions. A function app lets you group functions as a logical unit for easier management, deployment, scaling, and sharing of resources.

1. From the Azure portal menu or the **Home** page, select **Create a resource**.
2. In the **New** page, select **Compute** > **Function App**.
3. On the **Basics** page, use the function app settings
4. Select **Next : Hosting**. On the **Hosting** page, enter all the settings
5. Select **Next : Monitoring**. On the **Monitoring** page, enter the settings
6. Select **Review + create** to review the app configuration selections.
7. On the **Review + create** page, review your settings, and then select **Create** to provision and deploy the function app.
8. Select the **Notifications** icon in the upper-right corner of the portal and watch for the **Deployment succeeded** message.
9. Select **Go to resource** to view your new function app. You can also select **Pin to dashboard**. Pinning makes it easier to return to this function app resource from your dashboard.

Next, create a function in the new function app.

Create an HTTP trigger function

1. From the left menu of the **Function App** window, select **Functions**, and then select **Create** from the top menu.
2. From the **Create Function** window, leave the **Development environment** property as **Develop in portal**, and then select the **HTTP trigger** template.
3. Under **Template details** use HttpExample for **New Function**, select **Anonymous** from the **Authorization level** drop-down list, and then select **Create**.

   Azure creates the HTTP trigger function. Now, you can run the new function by sending an HTTP request.

References:

1. **https://www.tutorialride.com/cloud-computing/service-models-in-cloud-computing.htm**
2. **https://www.javatpoint.com/cloud-service-models**
3. **https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html#concepts-availability-zones**
4. **https://www.geeksforgeeks.org/cloud-deployment-models/**
5. **https://www.simplilearn.com/tutorials/aws-tutorial/what-is-aws**
6. **https://www.techtarget.com/searchaws/definition/AWS-bucket**
7. **https://k21academy.com/docker-kubernetes/kubernetes-architecture-components-overview-for-beginners/**
8. **http://ijcsit.com/docs/Volume%207/vol7issue3/ijcsit2016070326.pdf**
9. **https://learn.microsoft.com/en-us/azure/dns/private-dns-overview**
10. **Notes from Karnataka LMS.**
11. **https://learn.microsoft.com/en-us/azure/azure-functions/functions-create-function-app-portal**

Certificate

Certified that all the model answers are preparedby me for the subject/Course- Cloud Computing (20CS53I) and class – 5^{5h} sem and programme –Computer Science and Engineering and typed by me only. All anwers are within the prescribed syllabus and model answers and scheme of valuation prepared by me are correct.

DODDAMANI BASAVARAJ   Signature of paper setter
Lecturer, Dept Of CSE
Government Polytechnic for Women , Hubli