Assignment : Cloud Computing

**Submitted By :- Purv Patel**

**Module 5: Business Continuity In The Cloud**

**1-How to configure, develop and maintain Security and Privacy in cloud?**

### > **How to Configure, Develop, and Maintain Security & Privacy in Cloud Computing**

Cloud security and privacy are critical to protecting **data, applications, and infrastructure** from cyber threats. This involves **proper configuration, security development, and continuous maintenance**.

## **1️ Configuring Cloud Security & Privacy**

Proper **initial setup and configuration** ensures strong security in the cloud.

### **🔹 Steps for Secure Cloud Configuration:**

**Use Strong Identity & Access Management (IAM)**

* Apply **Role-Based Access Control (RBAC)**.
* Enable **Multi-Factor Authentication (MFA)** for users.
* Limit **privileges** based on the least privilege principle.

**Encrypt Data at Rest & in Transit**

* Use **AES-256 encryption** for stored data.
* Implement **SSL/TLS encryption** for data transmission.
* Utilize **Cloud Key Management Services (KMS)** (AWS KMS, Azure Key Vault).

**Configure Firewalls & Network Security**

* Set up **cloud security groups** (AWS Security Groups, Azure NSG).
* Implement **Web Application Firewalls (WAF)** to prevent attacks.
* Use **Zero Trust Network Access (ZTNA)** for secure remote access.

**Enable Logging & Monitoring**

* Use **cloud-native security tools** (AWS CloudTrail, Azure Sentinel, Google Security Command Center).
* Monitor logs with **SIEM (Security Information and Event Management)** solutions.
* Set up **automated alerts for security threats**.

**Secure APIs & Endpoints**

* Apply **OAuth 2.0** and **API gateways** to protect APIs.
* Restrict API access with **IP whitelisting** and **rate limiting**.

**Implement Secure Backup & Disaster Recovery**

* Automate **cloud backups** with versioning.
* Enable **geo-redundancy** for disaster recovery.

## **2️ Developing Secure Cloud Environments**

Cloud security must be integrated during development to prevent vulnerabilities.

### **🔹 Steps for Secure Cloud Development:**

**Follow Secure Coding Practices**

* Use **DevSecOps** for continuous security integration.
* Perform **code reviews and vulnerability scans** (SAST, DAST tools).

**Apply Data Masking & Anonymization**

* Mask sensitive data in development environments.
* Use **tokenization** for added privacy.

**Use Secure Containers & Microservices**

* Deploy applications using **Docker & Kubernetes security best practices**.
* Apply **pod security policies** in Kubernetes.

**Adopt Zero Trust Security Model**

* **Verify every request** instead of assuming trust.
* Enforce **least privilege access** for all users and services.

**Regular Security Testing**

* Conduct **penetration testing** on cloud infrastructure.
* Run **automated vulnerability scans** for continuous monitoring.

## **3️ Maintaining Cloud Security & Privacy**

Continuous monitoring and updates help maintain cloud security over time.

### **🔹 Steps for Maintaining Cloud Security:**

**Apply Regular Security Patches & Updates**

* Keep cloud services and OS **up-to-date**.
* Automate patch management with **AWS Patch Manager, Azure Update Management**.

**Monitor & Respond to Security Incidents**

* Use **Security Information and Event Management (SIEM)** solutions.
* Enable **automated threat response** with AI-driven security tools.

**Review & Audit Security Policies Regularly**

* Conduct **compliance audits** (GDPR, HIPAA, ISO 27001).
* Perform **cloud security posture assessments**.

**User Awareness & Training**

* Educate employees about **phishing, social engineering, and password security**.
* Conduct **regular security drills & incident response simulations**.

### **🔹 Conclusion**

🔹 **Configuring Cloud Security** → Set up IAM, encryption, firewalls, and monitoring.  
🔹 **Developing Secure Cloud Systems** → Follow secure coding, zero trust, and vulnerability testing.  
🔹 **Maintaining Security & Privacy** → Apply updates, monitor threats, and train employees.

**2-What is Portability in cloud?**

### > **What is Portability in Cloud Computing?**

**Portability in cloud computing** refers to the ability to **move applications, data, and services** between different cloud environments **with minimal changes or disruptions**. It ensures flexibility, reduces vendor lock-in, and improves operational efficiency.

## **🔹 Types of Cloud Portability**

### **1️ Data Portability**

Ability to transfer **data** between cloud providers without **loss or corruption**.  
 **Example:** Moving data from **AWS S3 to Google Cloud Storage**.

### **2️ Application Portability**

Running an application **on different cloud platforms** without major modifications.  
 **Example:** Deploying a containerized app on **AWS, Azure, and Google Cloud**.

### **3️ Service Portability**

Shifting entire cloud **services/workloads** between providers.  
 **Example:** Migrating a **database service from AWS RDS to Azure SQL Database**.

### **4️ Platform Portability**

Running applications across **on-premises, private, and public clouds**.  
 **Example:** Using **Kubernetes** to deploy apps on **AWS, Azure, or On-Prem**.

## **🔹 Challenges of Cloud Portability**

**Differences in cloud architectures** (e.g., AWS vs. Azure).  
 **Proprietary services** causing vendor lock-in.  
 **Data transfer complexities** (format, size, compliance).  
 **Security & compliance issues** when moving workloads.

**3-What is Reliability and high Availability in cloud?**

### > **Reliability and High Availability in Cloud Computing**

**Reliability** and **High Availability (HA)** are critical for ensuring **continuous operation, minimal downtime, and fault tolerance** in cloud environments.

## **🔹 1. Reliability in Cloud Computing**

**Reliability** refers to the ability of a cloud system to **consistently perform as expected** over time, without failures. It ensures that applications and services run **without frequent interruptions**.

### **✅ Key Features of Cloud Reliability:**

**Fault Tolerance** → The system can handle failures without affecting performance.  
 **Redundancy** → Data is replicated across multiple locations.  
 **Disaster Recovery** → Backups and failover mechanisms ensure quick recovery.  
 **Load Balancing** → Distributes traffic to prevent system overload.

**Example:**

* **Amazon S3** offers **99.999999999% (11 nines) durability**, ensuring data is highly reliable.

## **🔹 2. High Availability (HA) in Cloud Computing**

**High Availability (HA)** ensures that cloud services are **accessible and operational with minimal downtime**, even during failures.

### **✅ Key Features of High Availability:**

**Multi-Region Deployment** → Applications are deployed in multiple data centers.  
 **Failover Systems** → Automatic switching to backup servers in case of failure.  
 **Auto Scaling** → Resources adjust dynamically based on demand.  
 **Load Balancing** → Spreads traffic across multiple instances to prevent failure.

**Example:**

* **AWS Elastic Load Balancer (ELB)** distributes incoming traffic to multiple servers, ensuring **high availability**.

## **🔹 Difference Between Reliability & High Availability**

| Feature | Reliability | High Availability |
| --- | --- | --- |
| **Focus** | Prevents **failures** | Ensures **continuous uptime** |
| **Key Elements** | Fault tolerance, redundancy, backups | Failover, load balancing, auto-scaling |
| **Goal** | Ensures data integrity and system stability | Ensures systems remain accessible 24/7 |
| **Example** | Reliable cloud storage (AWS S3, Google Cloud Storage) | Cloud services with 99.99% uptime (AWS EC2, Azure VMs) |

### **🔹 How to Achieve High Reliability & Availability?**

**Deploy applications across multiple cloud regions.**  
 **Use load balancers to distribute traffic.**  
 **Implement automated failover systems.**  
 **Regularly test backup and disaster recovery plans.**  
 **Monitor cloud performance using observability tools (AWS CloudWatch, Azure Monitor).**

**4-Describe Mobility Cloud Computing**

### > **Mobility in Cloud Computing**

**Mobility in cloud computing** refers to the ability to access cloud-based applications, data, and services **anytime, anywhere, and from any device** with an internet connection. It enables **seamless computing experiences** across multiple platforms, improving flexibility and productivity.

## **🔹 Key Features of Mobility in Cloud Computing**

**Anywhere Access** → Users can access data and applications remotely via the internet.  
 **Multi-Device Compatibility** → Supports smartphones, tablets, laptops, and IoT devices.  
 **Real-Time Synchronization** → Changes are updated instantly across all connected devices.  
 **Scalability & Flexibility** → Users can scale resources as needed without hardware constraints.  
 **Security & Data Protection** → Uses authentication (MFA, IAM), encryption, and secure access controls.

**Example:**

* **Google Drive & Microsoft OneDrive** allow users to access files from any device.
* **Cloud-based CRM (e.g., Salesforce)** enables sales teams to work remotely.

## **🔹 Benefits of Mobility in Cloud Computing**

**Increased Productivity** → Employees can work remotely without location restrictions.  
 **Cost-Effective** → Reduces the need for on-premises infrastructure.  
 **Improved Collaboration** → Teams can share and edit documents in real-time.  
 **Disaster Recovery** → Data is stored securely in the cloud, preventing data loss.

## **🔹 Challenges of Cloud Mobility**

**Security Risks** → Increased risk of cyber threats (e.g., unauthorized access, data breaches).  
 **Network Dependency** → Requires a stable internet connection for optimal performance.  
 **Device Compatibility Issues** → Some applications may not work across all devices.

## **🔹 How to Enhance Mobility in Cloud Computing?**

Use **VPNs & Secure Access Controls** for safe remote access.  
 Implement **Zero Trust Security & Multi-Factor Authentication (MFA)**.  
 Ensure **cloud applications are optimized for mobile devices**.  
 Enable **offline access & automatic synchronization**.

**5-Describe AWS, Azure, Google cloud Platforms**

### > **Comparison of AWS, Azure, and Google Cloud Platforms (GCP)**

AWS, Azure, and Google Cloud are the **three leading cloud service providers**, offering a wide range of services like **computing, storage, networking, AI, and security** for businesses and developers.

## **🔹 1. Amazon Web Services (AWS)**

**Launched:** 2006  
 **Market Share:** ~32% (Largest cloud provider)  
 **Strengths:**

* Broadest range of services & global coverage.
* Strong in **IaaS (Infrastructure as a Service)**.
* Large ecosystem with **EC2 (Compute), S3 (Storage), and Lambda (Serverless Computing)**.  
   **Use Cases:** Startups, Enterprises, AI/ML workloads, Big Data, Hosting.  
   **Example:** Netflix, Airbnb, NASA, Samsung use AWS.

## **🔹 2. Microsoft Azure**

**Launched:** 2010  
 **Market Share:** ~22% (2nd Largest)  
 **Strengths:**

* Best for **Hybrid Cloud** (On-Prem + Cloud).
* Strong **Enterprise & Windows Integration** (Works with Active Directory, Office 365).
* Powerful AI/ML & analytics services.  
   **Use Cases:** Enterprises, Windows-based workloads, Hybrid Cloud, AI/ML.  
   **Example:** LinkedIn, BMW, Adobe, HP use Azure.

## **🔹 3. Google Cloud Platform (GCP)**

**Launched:** 2008  
 **Market Share:** ~11% (3rd Largest)  
 **Strengths:**

* Best for **AI/ML, Big Data, and Analytics**.
* Strong in **Kubernetes & Container-based deployments**.
* Cost-effective for startups & developers.  
   **Use Cases:** AI/ML, Data Analytics, App Development, Cloud-Native Apps.  
   **Example:** Spotify, PayPal, Twitter, HSBC use GCP.

## **🔹 Feature Comparison Table**

| Feature | AWS | Azure | Google Cloud |
| --- | --- | --- | --- |
| **Compute** | EC2, Lambda | Virtual Machines (VMs) | Compute Engine, Kubernetes |
| **Storage** | S3, EBS, Glacier | Blob, Disk, File Storage | Cloud Storage, Persistent Disks |
| **Networking** | VPC, CloudFront | Virtual Network, Load Balancer | VPC, Cloud CDN |
| **AI/ML** | SageMaker | Azure AI, ML Studio | TensorFlow, AI Hub |
| **Database** | RDS, DynamoDB | SQL DB, Cosmos DB | BigQuery, Firestore |
| **Hybrid Cloud** | VMware on AWS | Azure Arc, Hybrid Cloud | Anthos |
| **Pricing** | Pay-as-you-go, Free tier | Pay-as-you-go, Reserved | Flexible pricing |

## **🔹 Which One to Choose?**

**AWS:** Best for enterprises needing a mature, all-in-one cloud.  
 **Azure:** Best for Microsoft users, hybrid cloud, and enterprise solutions.  
 **GCP:** Best for AI/ML, analytics, and cost-effective cloud computing.

**6-Accessing AWS, Azure and Google cloud Platforms (any one portal )**

### > **How to Access AWS, Azure, or Google Cloud Platform (GCP) Portals**

You can access any cloud provider's **management portal** via a web browser. Below are the steps for **AWS**, **Azure**, and **Google Cloud (GCP).**

## **🔹 Accessing AWS Management Console**

### **Steps:**

1️ Go to [AWS Console](https://aws.amazon.com/console/)  
2️ Click **"Sign in to the Console"**  
3️ Enter your **AWS account email & password**  
4️ Click **"Sign In"**

Once logged in, you can access AWS services like **EC2, S3, Lambda, RDS, and IAM**.

**AWS CLI Access (Command Line)**

* Install AWS CLI: [Download Here](https://aws.amazon.com/cli/)
* Configure CLI: aws configure
* Run commands: aws s3 ls (List S3 Buckets)

## **🔹 Accessing Microsoft Azure Portal**

### **Steps:**

1️ Go to [Azure Portal](https://portal.azure.com/)  
2️ Click **"Sign In"**  
3️ Enter your **Microsoft credentials**  
4️ Click **"Next"** to access the dashboard

Once inside, you can manage **Virtual Machines, Azure AD, Storage, and Networking.**

**Azure CLI Access (Command Line)**

* Install Azure CLI: [Download Here](https://docs.microsoft.com/en-us/cli/azure/install-azure-cli)
* Login: az login
* Run commands: az vm list (List Virtual Machines)

## **🔹 Accessing Google Cloud Platform (GCP) Console**

### **Steps:**

1️ Go to [Google Cloud Console](https://console.cloud.google.com/)  
2️ Click **"Sign In"**  
3️ Log in with your **Google account**  
4️ Access **Compute Engine, Cloud Storage, BigQuery, and IAM.**

**GCP CLI Access (Command Line)**

* Install GCloud SDK: [Download Here](https://cloud.google.com/sdk/docs/install)
* Login: gcloud auth login
* Run commands: gcloud compute instances list (List Compute Instances)

### **Summary**

| Cloud | Portal URL | CLI Tool |
| --- | --- | --- |
| **AWS** | [AWS Console](https://aws.amazon.com/console/) | AWS CLI (aws configure) |
| **Azure** | [Azure Portal](https://portal.azure.com/) | Azure CLI (az login) |
| **GCP** | [Google Cloud Console](https://console.cloud.google.com/) | GCloud CLI (gcloud auth login) |

**7-Create compute, create network, create storage on AWS , Azure and GCP**

### > **How to Create Compute, Network, and Storage in AWS, Azure, and GCP**

Below are the steps to create **Compute (VMs), Networking (VPC), and Storage (Buckets/Disks)** on **AWS, Azure, and GCP.**

## **🔹 AWS (Amazon Web Services)**

### **1️ Create a Compute Instance (EC2)**

1️ Login to [AWS Console](https://aws.amazon.com/console/)  
2️ Go to **EC2 Dashboard** → Click **Launch Instance**  
3️ Select **Amazon Machine Image (AMI)** (e.g., Ubuntu, Windows)  
4️ Choose **Instance Type** (e.g., t2.micro for free tier)  
5️ Configure **Networking & Security Groups**  
6️ Add **Storage (EBS Volume)**  
7️ Click **Launch** and select a **key pair**

**CLI Command:**

aws ec2 run-instances --image-id ami-12345678 --instance-type t2.micro --key-name my-key --security-groups my-security-group

### **2️ Create a Network (VPC & Subnet)**

1️ Go to **VPC Dashboard** → Click **Create VPC**  
2️ Define **CIDR Block** (e.g., 10.0.0.0/16)  
3️ Click **Create Subnet** → Assign it to the VPC  
4️ Set up an **Internet Gateway (IGW)** for external access  
5️ Configure **Route Table** to allow traffic

**CLI Command:**

aws ec2 create-vpc --cidr-block 10.0.0.0/16

aws ec2 create-subnet --vpc-id vpc-12345678 --cidr-block 10.0.1.0/24

### **3️ Create a Storage Bucket (S3)**

1️ Go to **S3 Dashboard** → Click **Create Bucket**  
2️ Enter **Bucket Name** (must be unique)  
3️ Select **Region** (e.g., US East)  
4️ Configure **Public Access & Permissions**  
5️ Click **Create Bucket**

**CLI Command:**

aws s3 mb s3://my-bucket-name

## **🔹 Microsoft Azure**

### **1️ Create a Compute Instance (VM)**

1️ Login to [Azure Portal](https://portal.azure.com/)  
2️ Go to **Virtual Machines** → Click **Create VM**  
3️ Select **Image** (e.g., Ubuntu, Windows)  
4️ Choose **Size (B1s, D2s\_v3, etc.)**  
5️ Configure **Networking (VNet, NSG, Public IP)**  
6️ Attach **Storage (Managed Disks)**  
7️ Click **Review + Create**

**CLI Command:**

az vm create --resource-group myGroup --name myVM --image UbuntuLTS --size Standard\_B1s --admin-username azureuser --generate-ssh-keys

### **2️ Create a Network (VNet & Subnet)**

1️ Go to **Virtual Networks** → Click **Create Virtual Network**  
2️ Define **Address Space (10.0.0.0/16)**  
3️ Click **Create Subnet** (e.g., 10.0.1.0/24)  
4️ Set up **Network Security Groups (NSG)**  
5️ Click **Create**

**CLI Command:**

az network vnet create --name myVNet --resource-group myGroup --address-prefix 10.0.0.0/16

az network vnet subnet create --vnet-name myVNet --resource-group myGroup --name mySubnet --address-prefix 10.0.1.0/24

### **3️ Create a Storage Account (Blob Storage)**

1️ Go to **Storage Accounts** → Click **Create Storage Account**  
2️ Set **Account Name & Region**  
3️ Select **Storage Type (Standard/ Premium)**  
4️ Choose **Blob, File, or Disk Storage**  
5️ Click **Review + Create**

**CLI Command:**

az storage account create --name mystorageaccount --resource-group myGroup --location eastus --sku Standard\_LRS

## **🔹 Google Cloud Platform (GCP)**

### **1️ Create a Compute Instance (VM)**

1️ Login to [Google Cloud Console](https://console.cloud.google.com/)  
2️ Go to **Compute Engine** → Click **Create Instance**  
3️ Select **Machine Type (e2-micro, n1-standard-2, etc.)**  
4️ Choose **OS Image (Ubuntu, Windows, etc.)**  
5️ Configure **Firewall & Networking**  
6️ Click **Create**

**CLI Command:**

gcloud compute instances create my-vm --machine-type=e2-micro --image=ubuntu-2004-focal-v20220406 --zone=us-central1-a

### **2️ Create a Network (VPC & Subnet)**

1️ Go to **VPC Network** → Click **Create VPC**  
2️ Set **CIDR Range (e.g., 10.1.0.0/16)**  
3️ Add **Subnet & Assign IP Range (10.1.1.0/24)**  
4️ Enable **Firewall Rules (Allow HTTP, SSH, etc.)**  
5️ Click **Create**

**CLI Command:**

gcloud compute networks create my-vpc --subnet-mode=custom

gcloud compute networks subnets create my-subnet --network=my-vpc --range=10.1.1.0/24

### **3️ Create a Storage Bucket (Cloud Storage)**

1️ Go to **Cloud Storage** → Click **Create Bucket**  
2️ Set **Bucket Name & Location**  
3️ Choose **Storage Class (Standard, Nearline, Coldline)**  
4️ Configure **Access Control & Encryption**  
5️ Click **Create**

**CLI Command:**

gcloud storage buckets create my-bucket --location=us-central1

## **Summary Table**

| **Service** | **AWS** | **Azure** | **GCP** |
| --- | --- | --- | --- |
| **Compute (VM)** | **EC2** | **Azure VM** | **Compute Engine** |
| **Network (VPC & Subnet)** | **VPC & Subnet** | **Virtual Network (VNet)** | **VPC & Subnet** |
| **Storage (Bucket/Disk)** | **S3 (Object Storage)** | **Azure Blob Storage** | **Cloud Storage** |
| **CLI Command (Compute)** | aws ec2 run-instances | az vm create | gcloud compute instances create |

**8-Compare Cloud pricing of resources and services on all platform Amazon Web Services (AWS):**

> When comparing cloud pricing among Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), it's essential to consider various factors such as compute, storage, and networking costs. Below is a comparative overview based on recent data:

**1. Compute Pricing**

* **AWS**: Offers On-Demand EC2 instances priced per hour. For example, a t4g.xlarge instance with 4 vCPUs and 16 GB RAM is priced at $0.1344 per hour.
* **Azure**: Provides Virtual Machines with per-minute billing. A comparable B4ms instance with 4 vCPUs and 16 GB RAM costs approximately $0.166 per hour.
* **GCP**: Utilizes per-second billing for its Compute Engine instances. An e2-standard-4 instance with 4 vCPUs and 16 GB RAM is priced at around $0.150924 per hour.

**2. Storage Pricing**

* **AWS**: Charges $0.023 per GB per month for its S3 Standard storage.
* **Azure**: Offers Blob Storage at $0.021 per GB per month, making it slightly more cost-effective for storage needs.
* **GCP**: Prices its Cloud Storage at $0.023 per GB per month, aligning closely with AWS's storage pricing.

**3. Reserved Instances and Commitments**

All three providers offer discounts for long-term commitments:

* **AWS**: Provides Reserved Instances with up to a 41% discount for a 1-year commitment.
* **Azure**: Offers Reserved Savings with similar discounts, approximately 41% for a 1-year term.
* **GCP**: Features Committed Use Contracts, offering up to a 63% discount for a 1-year commitment, making it advantageous for sustained workloads.

**4. Additional Considerations**

* **Billing Models**: AWS and Azure primarily use per-hour billing, while GCP offers per-second billing, allowing for more precise cost management.
* **Data Transfer Costs**: Each provider has its own pricing for data egress, which can significantly impact costs depending on usage patterns.
* **Service Availability**: The range of services and their availability can vary by region and provider, potentially influencing the overall cost and suitability for specific workloads.