CLOUD SECURITY FUNDAMENTALS

WITH AWS, AZURE, AND GOOGLE CLOUD

SkillWeed

CONTENTS

MODULE 1: INTRODUCTION TO CLOUD SECURITY

LEARNING OBJECTIVES:

By the end of this module, learners will be able to:

- Understand the fundamentals of cloud computing and cloud security.
- Describe the cloud service models and deployment models.
- Understand the Shared Responsibility Model across major cloud providers.
- Identify key cloud security challenges and mitigation strategies.

1.1 WHAT IS CLOUD COMPUTING?

Cloud computing refers to the on-demand delivery of IT resources over the internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, organizations can access technology services such as computing power, storage, and databases from a cloud provider.

Types of Cloud Services (Service Models):

- **IaaS (Infrastructure as a Service):** Provides virtualized computing resources. *Examples: AWS EC2, Azure Virtual Machines, Google Compute Engine.*
- **PaaS (Platform as a Service):** Offers hardware and software tools over the internet.
  Examples: AWS Elastic Beanstalk, Azure App Services, Google App Engine.
- **SaaS (Software as a Service):** Delivers software applications over the internet. *Examples: Microsoft 365, Google Workspace, Salesforce.*

1.2 CLOUD DEPLOYMENT MODELS:

- **Public Cloud:** Services offered over the public internet.
  Examples: AWS, Azure, Google Cloud.
- **Private Cloud:** Services maintained on a private network.
  Examples: VMware-based private cloud, on-prem cloud environments.
- **Hybrid Cloud:** Combination of public and private cloud infrastructures.
  Example: Azure Stack, AWS Outposts.
- **Multi-cloud:** Using multiple cloud services from different vendors.
  Example: Using AWS for storage and Google Cloud for ML.

1.3 WHY CLOUD SECURITY IS CRITICAL

Cloud security involves the protection of data, applications, and infrastructures involved in cloud computing.

Core Areas of Cloud Security:

- **Data Protection:** Ensuring confidentiality, integrity, and availability of data.
- **Identity and Access Management (IAM):** Secure access control.
- **Governance and Compliance:** Regulatory requirements (e.g., HIPAA, GDPR, PCI-DSS).
- **Threat Detection and Response:** Identifying and mitigating attacks in real-time.
- **Resilience and Availability:** Backups, redundancy, disaster recovery.

1.4 THE SHARED RESPONSIBILITY MODEL

Each cloud provider operates under the **Shared Responsibility Model**.
This model defines what the **provider is responsible for** and what the **customer is responsible for**.

Breakdown by Provider:

- **AWS:**
  - *AWS Responsibility:* Security **of** the cloud (hardware, software, networking).
  - *Customer Responsibility:* Security **in** the cloud (data, identity, access).
- **Azure:**
  - *Microsoft Responsibility:* Physical infrastructure, foundational services.
  - *Customer Responsibility:* Data classification, access control, endpoint protection.
- **Google Cloud:**
  - *Google Responsibility:* Underlying cloud infrastructure, global network security.
  - *Customer Responsibility:* Secure configuration, IAM, encryption key management.

1.5 KEY CLOUD SECURITY CHALLENGES

1. **Misconfigurations:** Improper setup of cloud services (e.g., public S3 buckets).
2. **Data Breaches:** Unauthorized access to sensitive data.
3. **Lack of Visibility:** Limited insight into cloud resources and user actions.
4. **Insider Threats:** Malicious or negligent employees.
5. **Compliance Issues:** Failure to meet regulatory standards.

1.6 MITIGATION STRATEGIES

- Use automated tools for **configuration management**.
- Implement **least privilege access control** and MFA.
- Enable **logging and monitoring** (CloudTrail, Azure Monitor, Google Cloud Logs).
- Encrypt data **at rest** and **in transit**.
- Conduct regular **security assessments** and **penetration testing**.

1.7 REAL-WORLD USE CASE DISCUSSION

Scenario: A healthcare startup stores patient records in AWS S3 without encryption and publicly accessible.

- **Risk:** Violation of HIPAA, potential data breach.
- **Solution:** Enable server-side encryption, configure bucket policies, enable logging.

MODULE 2: IDENTITY AND ACCESS MANAGEMENT (IAM)

LEARNING OBJECTIVES

By the end of this module, learners will be able to:

- Understand the core principles of IAM in cloud environments.
- Configure users, groups, roles, and policies in AWS, Azure, and Google Cloud.
- Apply best practices for secure identity management and access control.
- Compare IAM features across the three major cloud platforms.

2.1 WHAT IS IAM?

Identity and Access Management (IAM) is the framework that enables organizations to manage digital identities and regulate access to cloud resources securely.

Core IAM Concepts:

- **Identity:** A digital representation of a user, application, or device.
- **Authentication:** Validating identity (e.g., passwords, MFA).
- **Authorization:** Granting appropriate access to resources.
- **Principle of Least Privilege:** Grant only the permissions necessary.
- **Separation of Duties:** Dividing roles to reduce risk of misuse.

2.2 IAM IN AWS

Core Components:

- **IAM Users:** Represent people or applications that interact with AWS.
- **IAM Groups:** Collection of users with shared permissions.
- **IAM Roles:** Assigned temporary credentials for services or users.
- **IAM Policies:** JSON documents defining permissions.

Security Features:

- **MFA (Multi-Factor Authentication)**
- **IAM Access Analyzer**
- **AWS Organizations for centralized account governance**

Best Practices:

- Do not use the root account for daily tasks.
- Apply permissions boundaries.
- Rotate credentials regularly.

Hands-On Task:

Create an IAM role in AWS with access to S3 and attach it to an EC2 instance.

2.3 IAM IN MICROSOFT AZURE (AZURE AD)

Core Components:

- **Azure Active Directory (Azure AD):** Microsoft's cloud-based identity service.
- **Users and Groups:** Managed through Azure AD or synced from on-prem AD.
- **Roles:** Built-in or custom roles assigned via RBAC (Role-Based Access Control).
- **Service Principals:** Identities for applications and services.
- **Managed Identities:** Automates identity management for Azure services.

Security Features:

- **Conditional Access Policies**
- **Privileged Identity Management (PIM)**
- **Identity Protection for risk-based policies**

Best Practices:

- Enable Conditional Access and MFA.
- Use PIM for just-in-time role access.
- Monitor sign-in logs and risk detections.

Hands-On Task: (Optional)

Assign a custom RBAC role to a user group for read-only access to Azure Storage.

2.4 IAM IN GOOGLE CLOUD

Core Components:

- **Identities:** Google accounts, service accounts, groups.
- **IAM Roles:** Primitive (Owner, Editor, Viewer), predefined, and custom roles.
- **IAM Policies:** Bindings of roles to members on resources.
- **Service Accounts:** Used by applications or VM instances.

Security Features:

- **IAM Recommender** for least-privilege suggestions.
- **Workforce Identity Federation**
- **Organization Policy Service** for centralized control

Best Practices:

- Use predefined roles over primitive ones.
- Use service accounts with limited scopes.
- Review IAM policy bindings regularly.

Hands-On Task:

Create a service account with access to Google Cloud Storage and bind it to a compute engine VM.

2.5 MULTI-CLOUD IAM COMPARISON TABLE

Feature	AWS	Azure	Google Cloud
User Identity	IAM User	Azure AD User	Google Account / Identity
Role/Access Control	IAM Role	RBAC Roles	IAM Roles
MFA Support	Yes (Virtual/Hardware)	Yes (Conditional Access)	Yes
Service Identity	IAM Role / Service Principal	Managed Identity / App Reg.	Service Account
Policy Language	JSON	JSON / Azure Portal GUI	IAM Policy Bindings (YAML/JSON)
Central Governance	AWS Organizations	Azure Management Groups	Resource Hierarchy (Org/Folders)

2.6 COMMON IAM SECURITY RISKS

- **Over-permissioned accounts**
- **Lack of role separation**
- **Credential leaks or reuse**
- **Inactive accounts left enabled**
- **Unmonitored access to critical assets**

2.7 MITIGATION AND BEST PRACTICES CHECKLIST

- Implement **least privilege** and **role-based access**.
- Enforce **MFA** for all users.
- Use **centralized identity management** tools.
- Regularly audit access logs and **rotate credentials**.
- Use **automation and provisioning tools** (e.g., AWS CloudFormation, Azure Bicep, GCP Deployment Manager).

MODULE 3: NETWORK SECURITY IN THE CLOUD

LEARNING OBJECTIVES:

By the end of this module, learners will be able to:

- Understand how cloud networking works.
- Design secure network architectures in cloud environments.
- Implement virtual firewalls, segmentation, and private connectivity.
- Compare and apply security tools and features across AWS, Azure, and GCP.

3.1 INTRODUCTION TO CLOUD NETWORKING

Cloud network security focuses on **protecting the integrity, confidentiality, and availability** of cloud-based network infrastructures.

Key Concepts:

- **Virtual Network (VNet/VPC):** A logically isolated network in the cloud.
- **Subnet:** Segment of an IP network.
- **Security Groups/Firewall Rules:** Control traffic in and out of instances.
- **Private IP vs Public IP:** Determines exposure to the internet.
- **Routing Tables and NAT:** Control traffic direction and internet access.

3.2 NETWORK SECURITY IN AWS

Key Components:

- **VPC (Virtual Private Cloud):** Customizable virtual network.
- **Subnets:** Public and private.
- **Route Tables:** Direct network traffic.
- **Internet Gateway:** Enables internet access.
- **NAT Gateway:** Allows outbound internet traffic for private subnets.
- **Security Groups:** Stateful firewalls for EC2 instances.
- **NACLs (Network ACLs):** Stateless firewalls at the subnet level.
- **VPC Peering:** Connects VPCs across regions or accounts.
- **AWS PrivateLink:** Private connectivity between VPCs and AWS services.

Best Practices:

- Use private subnets for backend services.
- Apply least privilege rules to security groups.
- Monitor with **VPC Flow Logs**.

3.3 NETWORK SECURITY IN MICROSOFT AZURE

Key Components:

- **Virtual Network (VNet):** Azure's private network space.
- **Subnets and Address Spaces**
- **Network Security Groups (NSGs):** Control traffic at subnet or NIC level.
- **Application Security Groups (ASGs):** Group similar workloads for simplified rule management.
- **Azure Firewall:** Stateful, centralized firewall.
- **Azure DDoS Protection:** Native mitigation for volumetric attacks.
- **ExpressRoute:** Private connection between on-prem and Azure.
- **Bastion Host:** Secure jump-box access without exposing VMs to public IPs.

Best Practices:

- Limit NSG rules and prioritize ASGs for easier management.
- Use **Bastion** or **Just-In-Time (JIT)** VM access.
- Enable **Diagnostics and NSG Flow Logs**.

3.4 NETWORK SECURITY IN GOOGLE CLOUD PLATFORM (GCP)

Key Components:

- **Virtual Private Cloud (VPC):** Global private network.
- **Subnets (Regional):** Custom or auto.
- **Firewall Rules:** Apply to resources using tags or service accounts.
- **Routes:** Control traffic direction.
- **Cloud NAT:** Enables outbound traffic from private instances.
- **Cloud Armor:** DDoS protection and WAF.
- **VPC Peering & Shared VPC:** For cross-project connectivity.
- **Private Service Connect:** Securely connect to Google APIs and third parties.

Best Practices:

- Use tags and service accounts for firewall rule targeting.
- Implement **Shared VPCs** for large org structures.
- Use **VPC Service Controls** for data exfiltration protection.

3.5 MULTI-CLOUD NETWORK SECURITY COMPARISON

Feature	AWS	Azure	Google Cloud
Private Network	VPC	VNet	VPC (Global)
Firewall	Security Groups, NACLs	NSG, Azure Firewall	Firewall Rules
DDoS Protection	AWS Shield	Azure DDoS Protection	Cloud Armor
Private Connectivity	AWS PrivateLink, VPN	ExpressRoute, VPN	Private Service Connect
Logging	VPC Flow Logs	NSG Flow Logs	VPC Flow Logs
Centralized Control	AWS Firewall Manager	Azure Policy	VPC Service Controls

3.6 COMMON NETWORK SECURITY THREATS

- **Open Ports and Misconfigured Firewalls**
- **Flat Network Topologies**
- **Lack of Segmentation**
- **Unencrypted Traffic**
- **Public Exposure of Private Resources**

3.7 MITIGATION STRATEGIES

- Use **zero-trust network access** principles.
- Segment workloads using **subnets and firewall rules**.
- Apply **least access policies** and **deny by default**.
- Encrypt all data in transit using **TLS/SSL**.
- Enable **intrusion detection and logging** for all traffic.

3.8 LAB EXERCISE IDEAS

1. **AWS:** Create a VPC with public/private subnets and secure it with SGs and NACLs.
2. **Azure:** Set up a VNet with NSGs and simulate access control using ASGs.
3. **Google Cloud:** Build a secure VPC, configure firewall rules, and use Cloud Armor for web protection.

MODULE 4:
DATA SECURITY IN THE CLOUD

LEARNING OBJECTIVES:

By the end of this module, learners will be able to:

- Understand the core principles of securing data in the cloud.
- Apply encryption and key management strategies.
- Protect data at rest and in transit across AWS, Azure, and Google Cloud.
- Implement data classification, masking, and loss prevention techniques.

4.1 INTRODUCTION TO DATA SECURITY

Cloud data security refers to practices, technologies, and policies that protect data across its lifecycle—**creation, storage, use, sharing, archiving, and deletion.**

Key Concepts:

- **Data at Rest:** Stored data (e.g., in storage buckets, databases, VMs)
- **Data in Transit:** Data moving between services, users, or devices
- **Data in Use:** Data being processed in memory
- **Encryption:** Protecting data using cryptographic techniques
- **Tokenization & Masking:** Obscuring sensitive information
- **Data Loss Prevention (DLP):** Identifying and preventing sensitive data exfiltration

4.2 DATA SECURITY IN AWS

Encryption Options:

- **Server-Side Encryption (SSE):**
    - *SSE-S3:* Managed by AWS
    - *SSE-KMS:* AWS Key Management Service
    - *SSE-C:* Customer-provided keys
- **Client-Side Encryption:** Encrypting data before uploading
- **KMS (Key Management Service):** Key creation, rotation, and management
- **Envelope Encryption:** Protecting data with data keys that are encrypted with master keys

Additional AWS Features:

- **S3 Bucket Policies:** Restrict access at the object and bucket level
- **S3 Object Lock & Versioning:** Prevent deletions or overwrites
- **Macie:** AI-powered data classification and DLP
- **AWS Backup:** Centralized backup and restore

Best Practices:

- Enable **S3 default encryption**
- Use **KMS with key policies**
- Apply **bucket policies** to restrict public access

4.3 DATA SECURITY IN MICROSOFT AZURE

Encryption Options:

- **Azure Storage Encryption (ASE):** Automatic encryption at rest
- **Azure Disk Encryption (ADE):** For VMs using BitLocker or DM-Crypt
- **Azure Key Vault:** Centralized key management and secrets storage
- **TLS/SSL:** Encrypted transmission using Azure-managed or custom certificates

Data Protection Features:

- **Azure Information Protection (AIP):** Classification, labeling, and protection of sensitive data
- **Microsoft Purview (formerly Azure Purview):** Data governance and compliance
- **DLP Policies via Microsoft Defender & Purview**
- **Immutable Storage:** Prevent modifications to blob storage

Best Practices:

- Integrate **Key Vault with applications**
- Use **Private Endpoints** to limit data exposure
- Set **retention and deletion** policies for data compliance

4.4 DATA SECURITY IN GOOGLE CLOUD (GCP)

Encryption Options:

- **Automatic Encryption at Rest:** Default for all storage services
- **Customer-Managed Encryption Keys (CMEK):** Via Cloud KMS
- **Customer-Supplied Encryption Keys (CSEK):** Bring your own keys
- **Envelope Encryption:** Similar to AWS

Data Protection Features:

- **Cloud DLP API:** Scans and de-identifies sensitive data using AI
- **Cloud Key Management Service (KMS):** Key lifecycle management
- **Confidential Computing:** Protects data in use with secure enclaves
- **VPC Service Controls:** Prevent data exfiltration from sensitive services

Best Practices:

- Use **Cloud DLP** to discover and classify sensitive data
- Apply **IAM policies** to control access to Cloud Storage
- Use **Audit Logs** to monitor data access

4.5 COMPARISON: CLOUD DATA SECURITY CAPABILITIES

Feature	AWS	Azure	Google Cloud
Default Encryption	Enabled at rest	Enabled at rest	Enabled at rest
Customer-Managed Keys	AWS KMS	Azure Key Vault	Cloud KMS
DLP Tool	Amazon Macie	Microsoft Purview / AIP	Cloud DLP API
Immutable Storage	S3 Object Lock	Immutable Blob Storage	Object Versioning
Data Classification	Macie, Tags	Purview, Sensitivity Labels	Cloud DLP, Resource Labels
Secure Transmission	TLS/SSL, HTTPS	TLS/SSL, Private Links	TLS/SSL, Private Service Conn

4.6 THREATS TO DATA IN THE CLOUD

- **Unauthorized Access**
- **Data Breaches**
- **Misconfigured Storage Buckets**
- **Key Exposure**
- **Shadow IT / Untracked Data Storage**
- **Unencrypted or Insecure Transmission**

4.7 MITIGATION STRATEGIES

- Apply **least privilege access** to all data assets
- Use **automated encryption** for all data layers
- Implement **logging and alerting** on sensitive data access
- Apply **DLP rules** to monitor for sensitive data transfers
- Conduct regular **security assessments and classification audits**

4.8 HANDS-ON LAB IDEAS (OPTIONAL)

1. **AWS Lab:** Configure S3 bucket encryption with KMS and enable Macie to scan for sensitive data.
2. **Azure Lab:** Create a Key Vault, store a secret, and link it to a VM for disk encryption.
3. **GCP Lab:** Use Cloud DLP to scan Cloud Storage buckets and anonymize personal data.

4.9 REVIEW & QUIZ

- **True/False:** Data is encrypted by default in all cloud platforms.
- **Multiple Choice:** Which service is used for key management in Azure?
- **Scenario:** Your company must prevent healthcare records from being downloaded to personal devices. Which tools should you use in each cloud platform?

MODULE 5: MONITORING, LOGGING, AND THREAT DETECTION

LEARNING OBJECTIVES:

By the end of this module, learners will be able to:

- Understand the importance of monitoring and logging in cloud environments.
- Identify cloud-native tools for threat detection.
- Implement log collection, alerting, and analysis in AWS, Azure, and Google Cloud.
- Use automation to detect and respond to cloud security incidents.

5.1 INTRODUCTION TO MONITORING AND THREAT DETECTION

Why It Matters:

- Logs are essential for **audit trails**, **incident response**, and **forensics**.
- Monitoring enables **proactive detection** of threats and **reactive analysis** after incidents.

Key Concepts:

- **Monitoring:** Continuous observation of system activities and performance.
- **Logging:** Systematically recording events for review.
- **Threat Detection:** Identifying malicious activity and vulnerabilities.
- **SIEM (Security Information and Event Management):** Centralized platform for log aggregation and analysis.

5.2 MONITORING AND LOGGING IN AWS

Core Services:

- **CloudTrail:** Logs all API calls across AWS services.
- **CloudWatch:**
  - *Metrics:* CPU usage, latency, etc.
  - *Logs:* Collect and analyze logs from EC2, Lambda, etc.
  - *Alarms:* Triggered based on thresholds.
- **VPC Flow Logs:** Captures IP traffic flow within the VPC.
- **AWS Config:** Tracks configuration changes and compliance.

Threat Detection Tools:

- **Amazon GuardDuty:** Threat intelligence-based detection of anomalies and malware.
- **AWS Security Hub:** Centralizes findings from multiple security services.
- **AWS Inspector:** Scans EC2 for software vulnerabilities and network exposure.

Best Practices:

- Enable CloudTrail across all regions.
- Use CloudWatch Logs Insights for log queries.
- Integrate GuardDuty findings into Security Hub for response workflows.

5.3 MONITORING AND LOGGING IN MICROSOFT AZURE

Core Services:

- **Azure Monitor:**
  - Metrics and diagnostics from Azure services.
  - Log Analytics Workspace for central log queries.
- **Azure Activity Logs:** Audit control-plane actions.
- **Azure Diagnostic Settings:** Route logs to storage, Event Hub, or Log Analytics.
- **Azure Metrics Explorer:** Visualize performance metrics.

Threat Detection Tools:

- **Microsoft Defender for Cloud:** Real-time threat detection, vulnerability management.
- **Azure Sentinel:** Cloud-native SIEM and SOAR platform.
- **Microsoft Defender for Endpoint / Identity / SQL:** Specialized protection.

Best Practices:

- Collect logs from all resources using Diagnostic Settings.
- Enable alerts for anomalies (e.g., failed logins, unexpected location logins).
- Use Kusto Query Language (KQL) to explore logs in Log Analytics.

5.4 MONITORING AND LOGGING IN GOOGLE CLOUD (GCP)

Core Services:

- **Cloud Audit Logs:** Tracks Admin, Data, and System access.
- **Cloud Logging (formerly Stackdriver):** Ingests logs from services and VMs.
- **Cloud Monitoring:** Monitors uptime, metrics, and creates dashboards.
- **Cloud Trace and Profiler:** Performance diagnostics.

Threat Detection Tools:

- **Security Command Center (SCC):** Central dashboard for risk insights.
- **Cloud IDS:** Intrusion detection system powered by Palo Alto Networks.
- **Cloud Armor:** WAF with threat protection rules.
- **Cloud Security Scanner:** Scans App Engine, GKE, Compute Engine for vulnerabilities.

Best Practices:

- Use Log Buckets for fine-grained access control.
- Set up alerting policies in Cloud Monitoring.
- Integrate SCC with third-party SIEMs for unified monitoring.

5.5 MULTI-CLOUD MONITORING COMPARISON

Feature / Tool	AWS	Azure	Google Cloud
API Logging	CloudTrail	Activity Logs	Cloud Audit Logs
Metrics & Performance	CloudWatch	Azure Monitor	Cloud Monitoring
Threat Detection	GuardDuty	Defender for Cloud	Security Command Center
SIEM Integration	Security Hub + EventBridge	Azure Sentinel	Chronicle / SCC Integration
Flow Logging	VPC Flow Logs	NSG Flow Logs	VPC Flow Logs
Vulnerability Scanning	AWS Inspector	Defender for Servers	Cloud Security Scanner
Automation	Lambda, EventBridge	Logic Apps, Playbooks	Cloud Functions, Workflows

5.6 COMMON MONITORING AND DETECTION CHALLENGES

- **Data overload** without prioritization
- **Lack of centralized visibility** across regions/accounts/projects
- **Delayed alerting** and response time
- **Inconsistent log retention policies**
- **Alert fatigue** from false positives

5.7 MITIGATION & BEST PRACTICES

- Enable and standardize logging across all environments.
- Define critical log sources: IAM, Storage, Network, Compute.
- Implement **automated alerts and response** for high-risk activities.
- Use **SIEM and SOAR tools** for correlation and orchestration.
- Regularly review and refine **detection rules and baselines**.

5.8 HANDS-ON LAB IDEAS

1. **AWS Lab:** Enable CloudTrail, CloudWatch Logs, and set up GuardDuty alerts.
2. **Azure Lab:** Connect Azure resources to Azure Monitor and visualize with KQL.
3. **GCP Lab:** Use Cloud Audit Logs and create alert policies in Monitoring.

5.9 REVIEW & QUIZ

- **Scenario:** An attacker accessed a GCP storage bucket from an unknown IP. Which logs should you check?
- **Multiple-Choice:** What does AWS GuardDuty detect?
- **Drag-and-Drop:** Match monitoring tools to the cloud platform.

MODULE 6: COMPLIANCE AND GOVERNANCE IN THE CLOUD

LEARNING OBJECTIVES:

By the end of this module, learners will be able to:

- Understand key compliance frameworks relevant to cloud environments.
- Apply governance principles and tools for risk management.
- Use native cloud services to ensure regulatory compliance and enforce policies.
- Build a compliance strategy using automation and continuous monitoring.

6.1 WHAT IS CLOUD COMPLIANCE AND GOVERNANCE?

Compliance:

Adherence to **laws, regulations, industry standards**, and organizational policies (e.g., HIPAA, GDPR, SOC 2, ISO 27001).

Governance:

Defines **who can do what** in the cloud, **how resources are managed**, and **how policies are enforced** to align with business and regulatory objectives.

6.2 COMMON REGULATORY FRAMEWORKS IN CLOUD SECURITY

- **HIPAA:** U.S. healthcare data privacy and security
- **GDPR:** European Union General Data Protection Regulation
- **PCI-DSS:** Payment Card Industry Data Security Standard
- **ISO/IEC 27001:** Information security management system (ISMS)
- **FedRAMP:** U.S. federal cloud security standard
- **SOC 2:** Trust principles for SaaS providers (security, availability, confidentiality, etc.)

6.3 COMPLIANCE AND GOVERNANCE IN AWS

Key Tools:

- **AWS Artifact:** Provides access to AWS compliance reports and certifications.
- **AWS Config:** Tracks configuration changes and evaluates against rules.
- **AWS Organizations:** Manages accounts with Service Control Policies (SCPs).
- **AWS Control Tower:** Enforces guardrails, sets up landing zones.
- **AWS Audit Manager:** Continuously assesses AWS environments against compliance frameworks.

Best Practices:

- Use **SCPs** to enforce baseline restrictions.
- Leverage **Config Rules** for real-time compliance checks.
- Automate audit evidence collection with **Audit Manager**.

6.4 COMPLIANCE AND GOVERNANCE IN MICROSOFT AZURE

Key Tools:

- **Microsoft Purview Compliance Manager:** Tracks regulatory compliance and control implementation.
- **Azure Policy:** Enforces rules on resources (e.g., location, type, configuration).
- **Azure Blueprints:** Deploy preconfigured environments aligned to compliance frameworks.
- **Azure Resource Graph:** Query resources for governance analysis.
- **Management Groups:** Apply policies across multiple subscriptions.

Best Practices:

- Assign **policies and initiatives** for continuous compliance.
- Use **Blueprints** for repeatable governance templates.
- Integrate with **Compliance Manager** to track progress against frameworks.

6.5 COMPLIANCE AND GOVERNANCE IN GOOGLE CLOUD (GCP)

Key Tools:

- **Compliance Reports:** Available in Google Cloud Console under Assured Workloads.
- **Organization Policy Service:** Enforce policies like allowed regions or resource types.

- **Assured Workloads:** Enables compliance-aligned projects (FedRAMP, HIPAA, CJIS).
- **Forseti Security:** Open-source toolkit for compliance auditing and monitoring.
- **Cloud Asset Inventory:** Tracks resource metadata across services.

Best Practices:

- Use **Assured Workloads** for regulated workloads.
- Apply **Org Policies** to restrict services, configurations, and geography.
- Automate audits using **Cloud Asset Inventory** and **Security Command Center**.

6.6 MULTI-CLOUD GOVERNANCE COMPARISON

Capability	AWS	Azure	Google Cloud
Policy Enforcement Tool	AWS Organizations + SCPs	Azure Policy + Blueprints	Organization Policy Service
Compliance Documentation	AWS Artifact	Microsoft Compliance Manager	Assured Workloads Reports
Config/Drift Management	AWS Config	Azure Resource Manager	Cloud Asset Inventory
Automated Auditing	Audit Manager	Compliance Scorecard	Forseti Security
Pre-built Governance Templates	Control Tower	Azure Blueprints	Assured Workloads

6.7 CHALLENGES IN CLOUD COMPLIANCE

- Managing **multi-region, multi-cloud** compliance
- Keeping up with **changing regulations**
- Ensuring **consistent policy enforcement**
- Proving compliance during **audits**
- Dealing with **shadow IT** or unmanaged accounts

6.8 MITIGATION STRATEGIES

- Implement **policy-as-code** for consistency (Terraform, ARM, Deployment Manager)
- Enable **continuous compliance monitoring** using cloud-native tools
- Perform **gap assessments** against required frameworks
- Automate **report generation** for auditors
- Train staff on cloud-specific **compliance responsibilities**

MODULE 7:
SECURE DEVOPS (DEVSECOPS) IN THE CLOUD

LEARNING OBJECTIVES:

By the end of this module, learners will be able to:

- Understand the concept of DevSecOps and its importance in cloud environments.
- Integrate security into all phases of the CI/CD pipeline.
- Identify DevSecOps tools and practices across AWS, Azure, and Google Cloud.
- Implement secure coding, testing, and deployment practices.

7.1 WHAT IS DEVSECOPS?

DevSecOps = Development + Security + Operations

A cultural and technical movement that integrates security into the DevOps process to ensure **continuous security** throughout the **software development lifecycle (SDLC)**.

Core Principles:

- **Shift Left Security:** Integrate security early in the development lifecycle.
- **Automation:** Use tools to scan code, dependencies, and configurations.
- **Collaboration:** Foster cooperation between developers, security, and operations.
- **Continuous Monitoring:** Detect vulnerabilities across the pipeline and infrastructure.

7.2 SECURE DEVOPS LIFECYCLE

Phase	Security Integration
Plan	Threat modeling, secure design review
Develop	Secure coding standards, secrets management
Build	Code analysis, dependency scanning
Test	SAST/DAST, unit/security testing
Release	Signatures, compliance gates
Deploy	Infrastructure as Code (IaC) security, container scanning
Operate	Monitoring, incident response, patching
Monitor	SIEM, anomaly detection, audit logging

7.3 DEVSECOPS IN AWS

Tools & Services:

- **CodePipeline / CodeBuild / CodeDeploy:** Native CI/CD automation
- **Amazon Inspector:** Automates security assessments on EC2, Lambda, containers
- **CodeGuru Reviewer:** ML-based code review for security issues
- **Secrets Manager / Parameter Store:** Secure storage of credentials
- **CloudFormation Guard:** Validates security best practices in IaC templates
- **ECR Image Scanning:** Scans Docker containers for vulnerabilities

Best Practices:

- Use IAM roles instead of hardcoded credentials
- Automate dependency scanning with **CodeBuild + Inspector**
- Implement pre-deployment security checks in **CodePipeline**

7.4 DEVSECOPS IN MICROSOFT AZURE

Tools & Services:

- **Azure DevOps Pipelines:** Automates build and release workflows
- **Microsoft Defender for DevOps:** Integrates security into pipelines
- **Azure Key Vault:** Manages secrets and encryption keys
- **Azure Policy as Code:** Enforces governance at deployment
- **GitHub Advanced Security:** For code scanning, secret detection, and dependabot alerts
- **Microsoft Defender for Containers:** Container image scanning, runtime protection

Best Practices:

- Integrate security gates in Azure Pipelines
- Store all secrets in **Key Vault** with RBAC
- Use **Infrastructure-as-Code scanning** with Terraform or Bicep linters

7.5 DEVSECOPS IN GOOGLE CLOUD (GCP)

Tools & Services:

- **Cloud Build:** CI/CD platform with built-in security features
- **Binary Authorization:** Ensures only trusted container images are deployed
- **Cloud Source Repositories / GitHub / GitLab:** SCM integration
- **Artifact Registry + Image Scanning:** Scans container images for vulnerabilities
- **Secret Manager:** Secure secrets storage
- **Policy Controller (OPA Gatekeeper):** Enforce policies in GKE deployments

Best Practices:

- Require signed artifacts before deployment
- Use **Binary Authorization** for GKE workloads
- Enable automatic scanning of containers in **Artifact Registry**

7.6 MULTI-CLOUD DEVSECOPS COMPARISON

Feature / Capability	AWS	Azure	Google Cloud
CI/CD Tool	CodePipeline, CodeBuild	Azure DevOps, GitHub Actions	Cloud Build
Secret Management	Secrets Manager, SSM	Azure Key Vault	Secret Manager
Container Scanning	Amazon Inspector, ECR Scan	Defender for Containers	Artifact Registry Scan
Infrastructure as Code (IaC)	CloudFormation, CDK	Bicep, ARM	Deployment Manager, Terraform
Policy Enforcement	Config, CloudFormation Guard	Azure Policy	Policy Controller (OPA)
Code Analysis	CodeGuru	GitHub Advanced Security	GitHub/Custom CI Scanners

7.7 KEY DEVSECOPS CHALLENGES

- **Tool integration complexity**
- **Developer resistance to added security**
- **Inconsistent environments across teams**
- **Lack of automated testing for security**
- **Secrets sprawl and mismanagement**

7.8 DEVSECOPS MITIGATION STRATEGIES

- Adopt **security as code** for consistent policies
- Automate **SAST**, **DAST**, and **container scans** in pipelines
- Train teams in **secure development practices**
- Use **immutable infrastructure** and **trusted registries**
- Integrate **SIEM** and alerts into DevOps dashboards

MODULE 8:
HANDS-ON LABS AND CAPSTONE PROJECT (OPTIONAL)

LEARNING OBJECTIVES:

By the end of this module, learners will be able to:

- Apply concepts learned throughout the course in practical lab scenarios.
- Configure and secure resources in AWS, Azure, and Google Cloud.
- Analyze real-world scenarios and implement cloud security solutions.
- Design and present a secure cloud architecture using multi-cloud services.

8.1 HANDS-ON LAB SERIES (OPTIONAL)

Each lab is designed to simulate real-world tasks and includes guided and challenge-based activities.

Lab 1: Identity and Access Management (IAM) (Optional)

Goal: Implement secure access controls using IAM in each cloud environment.

- **AWS:**
  - Create users, roles, and groups
  - Assign least privilege policies
  - Enable MFA
- **Azure:**
  - Create Azure AD roles and conditional access policies
  - Apply RBAC to a resource group
- **GCP:**
  - Create service accounts
  - Assign roles using IAM policies
  - Use audit logs to review access events

Lab 2: Network Security Configuration (Optional)

Goal: Build a secure 3-tier architecture with firewalls, private subnets, and flow logs.

- Set up:
  - Public and private subnets
  - VPC/Virtual Network
  - Security Groups / NSGs / Firewall Rules
  - NAT gateway or equivalent

Lab 3: Data Security and Encryption (Optional)

Goal: Encrypt storage resources and manage keys securely.

- **AWS:**
  - Enable S3 encryption using KMS
  - Rotate keys
  - Use IAM policies to restrict access
- **Azure:**
  - Use Key Vault to encrypt a storage account
  - Configure access policies
- **GCP:**
  - Encrypt a Cloud Storage bucket with CMEK
  - Use Cloud KMS to manage keys

Lab 4: Monitoring, Logging, and Threat Detection (Optional)

Goal: Enable threat detection and monitoring tools to detect security incidents.

- **AWS:**
  - Configure CloudTrail and GuardDuty
  - Set alerts in CloudWatch
- **Azure:**
  - Use Azure Monitor and Log Analytics
  - Enable Microsoft Defender for Cloud
- **GCP:**
  - Set up Cloud Audit Logs and SCC
  - Enable alert policies and email notifications

Lab 5: DevSecOps Pipeline (Optional)

Goal: Build a secure CI/CD pipeline that includes code scanning and secret management.

- Create a pipeline using:
    - AWS CodePipeline + CodeBuild + ECR Scan
    - Azure DevOps + GitHub Advanced Security + Key Vault
    - GCP Cloud Build + Binary Authorization + Artifact Registry

8.2 CAPSTONE PROJECT: SECURE MULTI-CLOUD APPLICATION DEPLOYMENT (OPTIONAL)

Scenario:

You have been hired as a cloud security architect for a startup building a healthcare analytics platform. The company wants to host services on **AWS (backend APIs)**, **Azure (databases)**, and **Google Cloud (machine learning & storage)**.

Project Requirements:

1. **Design a secure architecture** using all three platforms:
    - Use private subnets for sensitive data
    - Enforce encryption at rest and in transit
    - Use IAM roles for service-to-service communication
2. **Implement Governance & Compliance:**
    - Apply policies to restrict regions and resource types
    - Enable centralized logging and audit trails
    - Generate a HIPAA-aligned control mapping
3. **Secure the DevOps Pipeline:**
    - Automate testing and deployment
    - Scan code, containers, and configurations

4. **Threat Detection and Response Plan:**
   - Enable GuardDuty, Defender for Cloud, and SCC
   - Define response actions for unauthorized access or malware detection

Deliverables:

- **Architecture Diagram**
- **Configuration Scripts (IaC: Terraform, Bicep, or YAML)**
- **Security Policies and IAM Configuration**
- **Monitoring & Response Workflow**
- **Compliance Summary Report**

SkillWeed