

I. Create a Cloud Organization in AWS/Google Cloud/or any equivalent Open Source cloud softwares like Openstack, Eucalyptus, OpenNebula with Role-based access control

1. Cloud Security with Microsoft Azure Active Directory (AAD)"

Experiment Title: "RBAC Deployment for Cloud Security with Microsoft Azure Active Directory (AAD)"

Objective: Deploy RBAC in Azure AD to enhance cloud security, managing user access and permissions to Azure resources through role-based authorization.

Requirements:

- Access to Azure AD with appropriate administrative rights.
- Definition of role-based access control policies for cloud resources.
- Knowledge of Azure infrastructure and RBAC configurations.

2. Cloud Resource Access with Google Cloud IAM

Experiment Title: "RBAC Setup for Specific Cloud Resource Access with Google Cloud IAM"

Objective: Set up RBAC using Google Cloud Identity and Access Management (IAM) for granular control over cloud resource access, assigning detailed access privileges based on user roles.

Requirements:

- Access to Google Cloud Console with administrative rights.
- Definition of fine-grained access policies for cloud services.
- Knowledge of Google Cloud IAM's detailed role structures and policies.

3. Audit Trails in IBM Security ISIGI

Experiment Title: " Audit Trails in IBM Security Identity Governance and Intelligence (ISIGI)"

Objective: Integrate RBAC within ISIGI to maintain compliance, creating audit trails to track user access and ensure adherence to compliance standards.

Requirements:

- Access to IBM ISIGI management interface.
- Definition of RBAC policies for compliance management.
- Knowledge of audit trail configuration and compliance regulations.

4. Web Security with F5 BIG-IP APM

Experiment Title: "RBAC Setup for Web Security with F5 BIG-IP Access Policy Manager (APM)"

Objective: Set up RBAC using F5 BIG-IP APM to reinforce web security by controlling user access and permissions within web applications based on assigned roles.

Requirements:

- Access to F5 BIG-IP APM configuration interface.
- Definition of role-based policies for web application security.
- Understanding of web application architecture and access control settings.

II. Create a Cost-model for a web application using various services and do Cost-benefit analysis.

1. Cloud Service Cost Estimation and Allocation for Web Application

Experiment : "Cloud Service Cost Estimation and Allocation for Web Application."

Objective: To estimate and allocate costs for different cloud services (compute, storage, databases, etc.) required by a web application. Analyze strategies for accurate cost estimation and allocation to fit within the application's budget constraints.

Requirements:

- Access to a cloud service provider's cost estimation tool (e.g., AWS Simple Monthly Calculator, Azure Pricing Calculator, Google Cloud Pricing Calculator).
- Defined usage patterns and requirements for various services in the web application.
- Understanding of budget constraints for the application's cloud expenses.

2. Cloud Service Cost Estimation Methodologies for Web Applications

Experiment: "Cloud Service Cost Estimation Methodologies for Web Applications"

Objective: To explore and apply diverse methodologies for estimating costs linked to different cloud services utilized by web applications. Consider usage metrics, pricing models, and specific service characteristics to ensure accurate cost estimations.

Requirements:

- Access to multiple cloud service providers' pricing calculators (AWS, Azure, Google Cloud).

- Understanding of the web application's service requirements (compute, storage, databases, etc.).
 - Knowledge of the pricing structures and metrics offered by different cloud services.
3. **Cost Modeling for Web Applications**
Experiment Title: "Cost Modeling of Compute Services for Web Applications"
Objective: To create a cost model for compute services across different cloud providers (e.g., AWS EC2, Azure VMs, GCP Compute Engine) and perform a cost-benefit analysis to determine the most cost-effective option for hosting a web application.
Requirements:
 - Access to cost calculators or billing insights of compute services on AWS, Azure, and GCP.
 - Detailed information on compute resource requirements for the web application.
 - Metrics for performance and scalability for cost-benefit assessment.
 4. **Serverless Service Cost Comparison for Web Application Functions**
Experiment Title: "Serverless Service Cost Comparison for Web Application Functions"
Objective: Develop a cost model for serverless services (e.g., AWS Lambda, Azure Functions, GCP Cloud Functions) and perform a cost-benefit analysis to identify the most cost-effective solution for managing functions within a web application.
Requirements:
 - Access to cost calculators and usage insights for serverless services on AWS, Azure, and GCP.
 - Detailed information on function execution requirements for the web application.
 - Metrics for function runtime and scalability for cost assessment.

III. Create alerts for usage of Cloud resources

1. **Custom Alert Creation for Cloud Resource Usage**
Experiment: "Custom Alert Creation for Cloud Resource Usage"
Objective: To create custom alerts for specific cloud resources based on defined patterns and unique usage metrics.
Requirements:
 - Access to monitoring and alerting tools within the cloud environment.
 - Understanding of specific resource usage patterns and metrics.
 - Ability to customize alert conditions for different resources.
2. **Event-Based Cloud Resource Alerting**
Experiment: "Event-Based Cloud Resource Alerting"
Objective: To configure event-driven alerts for cloud resources, triggering notifications upon specific occurrences or anomalies in resource usage.
Requirements:

- Use of event-driven monitoring tools (e.g., Azure Event Grid, AWS EventBridge).
 - Defined conditions or patterns to trigger event-based alerts.
 - Access to set up alert configurations for different events.
3. **Threshold-based Monitoring and Notification Setup**
Experiment: "Threshold-based Monitoring and Notification Setup"
Objective: To establish threshold-based monitoring for critical cloud resources and set up notifications to prevent resource overutilization.
Requirements:
- Usage of a cloud service's monitoring tool (e.g., AWS CloudTrail, Azure Alerts).
 - Determination of critical resource thresholds.
 - Configuration access to set up notifications for identified thresholds
4. **Cross-Platform Resource Monitoring and Alerting**
Experiment: "Cross-Platform Resource Monitoring and Alerting"
Objective: To set up monitoring and alerting systems across hybrid or multi-cloud environments, ensuring consistent tracking and notification setups.
Requirements:
- Usage of cross-platform monitoring and management tools.
 - Understanding of monitoring variations across multiple cloud platforms.
 - Configuration capabilities for cross-platform notifications and alerts.

IV. Create Billing alerts for your Cloud Organization

1. **Cross-Platform Cost Alerts and Optimization**
Experiment Title: "Cross-Platform Cost Alerts and Optimization"
Objective: To establish cost alerts and optimization strategies across multi-cloud environments, ensuring uniform tracking and notification setup.
Requirements:
- Access to multi-cloud management tools or individual cloud service billing and notification features.
 - Understanding of cost variations in different cloud platforms.
 - Configuration capabilities for cross-platform cost alerts and optimization.
2. **Governance-based Cost Alert Setup in Cloud Consortia**
Experiment Title: "Governance-based Cost Alert Setup in Cloud Consortia"
Objective: To implement governance-based cost alerts within cloud consortia, ensuring regulatory compliance and unified cost notification strategies across consortium members.
Requirements:
- Utilization of cloud consortium management or individual cloud service billing and notification features.

- Understanding of governance-based cost control mechanisms within cloud consortia.
- Configuration capabilities for governance-centric cost alerts in cloud consortia.

3. Cost Notifications in Hybrid Cloud Environments

Experiment: " Cost Notifications in Hybrid Cloud Environments"

Objective: To implement cost notification strategies in hybrid cloud settings, balancing cost control measures across various integrated cloud environments.

Requirements:

- Usage of hybrid cloud management tools or individual cloud service billing and alert systems.
- Understanding of cost monitoring intricacies within hybrid cloud setups.
- Configuration access for optimized cost notifications across hybrid cloud platforms.

4. Cost Notifications for Financial Oversight in GCP

Experiment Title: "Configuring Cost Notifications for Financial Oversight in Google Cloud Platform (GCP)"

Objective: Set up billing alerts in Google Cloud Platform (GCP) to monitor expenses and receive notifications when costs approach or exceed specified thresholds, enabling better financial oversight and adherence to budget constraints.

Requirements:

- Access to the Google Cloud Console with appropriate administrative permissions.
- Determination of cost thresholds or predefined budget limits for effective cost monitoring.
- Understanding of GCP Billing and Cost Management tools to configure alerting mechanisms.

V. Compare Cloud cost for a simple web application across AWS, Azure and GCP and suggest the best one

1. Comparative Infrastructure Cost Analysis for a Simple Web App

Experiment Title: "Comparative Infrastructure Cost Analysis for a Simple Web App"

Objective: To compare infrastructure costs (compute, storage, network) for hosting a simple web application across AWS, Azure, and GCP to determine the most cost-effective option.

Requirements:

- Access to AWS, Azure, and GCP accounts for cost assessment.
- Specification of infrastructure needs for the web application.
- Understanding of cost models and pricing details for core services in each cloud platform.

2 . TCO Comparison for a Web Application

Experiment Title: "Total Cost of Ownership (TCO) Comparison for a Web Application"

Objective: Compare the total cost of ownership (TCO) of hosting a basic web application across AWS, Azure, and GCP, considering all related costs for an extended period.

Requirements:

- Access to TCO calculators or detailed cost breakdowns for AWS, Azure, and GCP.
- Detailed estimation of costs associated with resources, maintenance, and additional services over a defined period.
- Understanding of TCO elements in each cloud platform.

3 . Scalability Cost Analysis for a Growing Web Application

Experiment Title: "Scalability Cost Analysis for a Growing Web Application"

Objective: Assess the scalability costs in AWS, Azure, and GCP, examining cost implications as the web application scales in traffic and resource usage.

Requirements:

- Access to scaling and cost estimation tools in AWS, Azure, and GCP.
- Anticipation of traffic and resource needs for scaling the web application.
- Understanding of pricing adjustments and cost considerations for increased demand in each cloud provider.

4 . Cloud Provider Selection for Web Application

Experiment Title: "Cost-Effective Cloud Provider Selection for Web Application Hosting"

Objective: To identify the most cost-effective cloud provider among AWS, Azure, and GCP for hosting a basic web application by analyzing various service costs and features.

Requirements:

- Access to pricing information and cost calculators for services on AWS, Azure, and GCP.
- Comprehensive understanding of the web application's resource requirements.
- Evaluation criteria including performance metrics, scalability, and additional features.