### **CLOUD ASSIGNMENT - 1**

### 

### **Assignment 1: EC2 (Elastic Compute Cloud)**

1. **Launch an EC2 Instance**:
   * Launch an EC2 instance with the Amazon Linux AMI.
   * Select an instance type (t2.micro) and configure it for public access.
   * Document the instance ID and public IP address.
2. **Configure Security Group**:
   * Create a security group that allows SSH (port 22) and HTTP (port 80).
   * Attach the security group to your EC2 instance.
3. **Install and Configure a Web Server**:
   * Connect to your EC2 instance via SSH and install Apache or Nginx.
   * Verify that the web server is running by accessing the public IP address.
4. **Create a Bash Script**:
   * Write a script to automate the installation of the web server.
   * Document your script and its functionality.
5. **Task**:
   * Submit a report with screenshots of the EC2 instance and the web server running.

### **Assignment 2: IAM (Identity and Access Management)**

1. **Create IAM Users**:
   * Create three IAM users with different permissions (e.g., Admin, Read-Only, S3 Access).
   * Provide details of the policies attached to each user.
2. **Implement Multi-Factor Authentication (MFA)**:
   * Enable MFA for one of the IAM users.
   * Document the steps taken to set up MFA.
3. **Create IAM Roles**:
   * Create an IAM role for EC2 instances that allows access to S3 buckets.
   * Attach the role to your existing EC2 instance.
4. **Task**:
   * Write a short essay on the importance of IAM and best practices for managing access in AWS.

### **Assignment 3: S3 (Simple Storage Service)**

1. **Create and Configure an S3 Bucket**:
   * Create a new S3 bucket named yourname-s3-bucket.
   * Enable versioning and logging for the bucket.
2. **Bucket Policies and Permissions**:
   * Write a bucket policy that allows public access to specific files (e.g., images).
   * Apply this policy and test its functionality.
3. **Data Upload and Retrieval**:
   * Upload at least three different types of files (images, text documents, etc.) to your S3 bucket.
   * Use the AWS CLI to list the contents of your bucket.
4. **Task**:
   * Create a report detailing the configurations, policies, and commands used, including screenshots.

### **Assignment 4: Static Page Hosting Using S3**

1. **Static Website Hosting Setup**:
   * Create a new S3 bucket for hosting a static website.
   * Enable static website hosting and set the index document to index.html.
2. **Upload Website Files**:
   * Create a simple HTML page with your name and a brief bio.
   * Upload the HTML file to the S3 bucket.
3. **Test Website**:
   * Access the public URL of the static website.
   * Document the steps taken to configure and test the website.
4. **Task**:
   * Submit your HTML file along with a report that includes screenshots of the S3 configurations.

### **Assignment 5: VPC (Virtual Private Cloud)**

1. **Create a VPC**:
   * Create a new VPC with a CIDR block of your choice (e.g., 10.0.0.0/16).
   * Create two subnets: one public and one private.
2. **Set Up Route Tables**:
   * Configure route tables for both public and private subnets.
   * Associate the public subnet with an internet gateway.
3. **Launch an EC2 Instance in the VPC**:
   * Launch an EC2 instance in the public subnet.
   * Test connectivity from your local machine via SSH.
4. **Task**:
   * Create a network diagram illustrating your VPC architecture, including subnets and gateways.

### **Assignment 6: Integrating Services (EC2, S3, IAM)**

1. **Deploy a Web Application**:
   * Use an EC2 instance to deploy a web application that interacts with S3 (e.g., file upload/download functionality).
   * Ensure that the EC2 instance has the necessary IAM role to access S3.
2. **Implement Logging**:
   * Enable S3 server access logging for your bucket.
   * Document how to check the logs for file access.
3. **Task**:
   * Create a presentation (5-7 slides) summarizing your project, challenges faced, and solutions implemented.

### **Assignment 7: Security Best Practices in AWS**

1. **Identify Security Risks**:
   * List at least five potential security risks in an AWS environment, focusing on EC2, IAM, and S3.
2. **Best Practices**:
   * Research and document AWS best practices for securing EC2 instances, managing IAM permissions, and protecting S3 data.
3. **Implement a Security Solution**:
   * Choose one of the identified risks and propose a security solution or tool (e.g., AWS GuardDuty, AWS Config).
   * Provide a step-by-step guide on implementing this solution.
4. **Task**:
   * Submit a report that includes your risk analysis, proposed solutions, and implementation steps, along with references.

### **Assignment 8: EC2 (Elastic Compute Cloud)**

1. **Set Up an EC2 Instance with Custom AMI**:
   * Create a custom Amazon Machine Image (AMI) from an existing EC2 instance.
   * Launch a new EC2 instance using this custom AMI.
2. **Load Balancing with EC2**:
   * Set up an Elastic Load Balancer (ELB) to distribute traffic across multiple EC2 instances.
   * Document the configuration steps and test the load balancer by sending traffic to it.
3. **Task**:
   * Write a report explaining the benefits of using custom AMIs and load balancers, along with screenshots of your configurations.

### **Assignment 9: IAM (Identity and Access Management)**

1. **Create Groups and Attach Policies**:
   * Create two IAM groups: Developers and Admins.
   * Assign appropriate policies to each group based on the principle of least privilege.
2. **Permissions Boundary**:
   * Implement a permissions boundary for the Admins group to limit the maximum permissions available to its members.
   * Document the process of creating and attaching the boundary.
3. **Task**:
   * Provide a summary of how permissions boundaries enhance security in AWS, along with your configuration screenshots.

### **Assignment 10: S3 (Simple Storage Service)**

1. **S3 Bucket Versioning and Lifecycle Policies**:
   * Enable versioning for your S3 bucket and upload several versions of a file.
   * Create a lifecycle policy to delete old versions after a specified number of days.
2. **Cross-Region Replication**:
   * Set up cross-region replication between two S3 buckets (one in your primary region and one in a different region).
   * Document the setup process and test it by uploading files to the source bucket.
3. **Task**:
   * Submit a report detailing the versioning and replication configurations, along with relevant screenshots.

### **Assignment 11: Static Page Hosting Using S3**

1. **Custom Domain Hosting**:
   * Use Amazon Route 53 to register a domain name.
   * Configure the S3 bucket to serve a static website using your custom domain.
2. **SSL/TLS Configuration**:
   * Obtain an SSL certificate from AWS Certificate Manager (ACM) and configure your S3 bucket to serve content over HTTPS.
   * Document the steps taken to implement SSL.
3. **Task**:
   * Create a presentation explaining the process of custom domain hosting and SSL configuration, including challenges faced and how you overcame them.

### **Assignment 12: VPC (Virtual Private Cloud)**

1. **VPC Peering**:
   * Create a second VPC and set up VPC peering between your two VPCs.
   * Enable routing between the VPCs and test connectivity.
2. **NAT Gateway Configuration**:
   * Set up a NAT Gateway in your public subnet to allow instances in your private subnet to access the internet.
   * Document the steps taken and test the internet connectivity from a private EC2 instance.
3. **Task**:
   * Prepare a diagram showing your VPC peering and NAT Gateway setup, along with a report explaining the configurations.

### **Assignment 13: Integrating Services (EC2, S3, IAM)**

1. **Deploy a Serverless Application**:
   * Use AWS Lambda to create a serverless function that uploads files to your S3 bucket when triggered by an API Gateway.
   * Configure the necessary IAM roles for your Lambda function.
2. **Monitoring and Alerts**:
   * Set up CloudWatch alarms to monitor the performance of your Lambda function and S3 bucket.
   * Document the metrics being monitored and how alerts are configured.
3. **Task**:
   * Create a detailed report on your serverless architecture, including diagrams and explanations of how the components interact.

### **Assignment 14: Cost Management in AWS**

1. **Cost Explorer**:
   * Utilize AWS Cost Explorer to analyze your AWS spending over the last three months.
   * Identify areas where costs can be optimized.
2. **Budgets and Alerts**:
   * Create a budget for your AWS account and set up alerts for when your spending approaches the budget limit.
   * Document the steps taken to create the budget and the rationale behind the budget limit.
3. **Task**:
   * Submit a report detailing your cost analysis, optimization strategies, and budget configurations, including screenshots from the Cost Explorer.

### **Assignment 15: Security Best Practices in AWS**

1. **Conduct a Security Assessment**:
   * Perform a security assessment of your AWS environment using AWS Trusted Advisor and AWS Inspector.
   * Document the findings and recommend actions to improve security.
2. **Implement AWS Config**:
   * Set up AWS Config to monitor changes to your resources and ensure compliance with best practices.
   * Document the configuration steps and review the compliance dashboard.
3. **Task**:
   * Write a report summarizing your security assessment, recommendations for improvement, and the AWS Config setup, including relevant screenshots.

#### **Part 1: EC2 (Elastic Compute Cloud)**

1. **Setup an EC2 Instance**:
   * Launch an EC2 instance using the AWS Management Console.
   * Choose an appropriate Amazon Machine Image (AMI) and instance type.
   * Configure security groups to allow SSH access.
   * Note down the public IP address of your EC2 instance.
2. **Connect to the EC2 Instance**:
   * Use SSH to connect to your EC2 instance from your local machine.
   * Install a web server (e.g., Apache or Nginx) on your EC2 instance.
3. **Task**:
   * Document the steps you followed to launch and connect to the EC2 instance.
   * Include screenshots of your AWS Management Console and terminal.

#### **Part 2: IAM (Identity and Access Management)**

1. **Create IAM Users and Policies**:
   * Create a new IAM user with programmatic access.
   * Create a policy that grants the user permission to manage S3 buckets and attach it to the user.
2. **Task**:
   * Write a brief explanation of how IAM works in AWS and the importance of using IAM for security.
   * Include screenshots of the IAM user and policy creation process.

#### **Part 3: S3 (Simple Storage Service)**

1. **Create an S3 Bucket**:
   * Create a new S3 bucket using the AWS Management Console.
   * Enable versioning for the bucket.
   * Configure bucket policies to allow public read access for static website hosting.
2. **Static Page Hosting**:
   * Upload an HTML file (e.g., index.html) to your S3 bucket.
   * Configure the S3 bucket for static website hosting.
   * Note the endpoint URL for the hosted website.
3. **Task**:
   * Provide the HTML file you used for hosting.
   * Document the steps for configuring the S3 bucket and static website hosting, including any screenshots.

#### **Part 4: VPC (Virtual Private Cloud)**

1. **Create a VPC**:
   * Create a new VPC with public and private subnets.
   * Configure route tables and internet gateways to allow internet access for public subnets.
2. **Task**:
   * Explain the significance of using VPC in AWS.
   * Include a diagram of your VPC setup, labeling the subnets, route tables, and gateways.

#### **Submission Guidelines**

* Compile all documentation, screenshots, and code into a single PDF or Word document.

### **IAM Policy Simulations**

1. **IAM Policy Simulation**:
   * Create a scenario where a user needs to access an S3 bucket and an EC2 instance. Write an IAM policy that allows the user to perform the following actions:
     + List objects in the S3 bucket.
     + Launch, stop, and terminate EC2 instances.
   * Use the IAM Policy Simulator to test whether the policy works as intended and document the results.
2. **Policy Conditions**:
   * Write an IAM policy that allows a user to access an S3 bucket only from a specific IP address.
   * Explain the condition key used in the policy and test it using the IAM Policy Simulator.
3. **Deny Overrides Allow**:
   * Create an IAM policy that allows a user to access all EC2 actions but denies the ability to terminate instances.
   * Test this policy with the IAM Policy Simulator to demonstrate how deny rules take precedence over allow rules.
4. **Cross-Account Access**:
   * Write an IAM policy that grants a user in Account A access to an S3 bucket in Account B.
   * Explain the necessary configurations needed in both accounts to make this work.

### **S3 Bucket Versioning**

1. **Enable Versioning**:
   * Enable versioning on an existing S3 bucket and upload multiple versions of a file.
   * Document the steps taken and how to retrieve previous versions of the file using the AWS Management Console.
2. **Version Deletion**:
   * Create a scenario where you delete the latest version of a file in a versioned bucket.
   * Explain how to restore the deleted version and document the steps taken.
3. **S3 Inventory Reports**:
   * Configure S3 Inventory reports to track all versions of files in a versioned bucket.
   * Describe the steps to enable inventory reports and analyze the generated report.

### **S3 Lifecycle Policies**

1. **Lifecycle Policy Creation**:
   * Create a lifecycle policy that transitions objects to S3 Glacier after 30 days and deletes them after 365 days.
   * Document the policy creation process and explain how it helps manage storage costs.
2. **Testing Lifecycle Policies**:
   * Upload several objects to an S3 bucket and apply a lifecycle policy that deletes objects older than 7 days.
   * Test the policy and document the results by checking the bucket contents after the policy execution.
3. **Lifecycle Policy Logging**:
   * Set up logging to track changes made by lifecycle policies in your S3 bucket.
   * Explain how to review the logs to verify policy actions.

**EC2 Security Groups**:

* Create a security group that allows inbound traffic only from specific IP addresses for SSH and HTTP access.
* Document the process and test the security group settings by attempting to connect from allowed and disallowed IP addresses.

**EC2 Tags**:

* Launch multiple EC2 instances with different tags (e.g., environment, department).
* Write a script that lists all instances by their tags using the AWS CLI.

**EC2 Instance Types**:

* Research different EC2 instance types and their use cases. Create a comparison chart highlighting their CPU, memory, storage, and network performance.
* Write a recommendation on which instance type to use for a web application and justify your choice.

**Auto Scaling Groups**:

* Set up an Auto Scaling group with a minimum of 2 instances and a maximum of 5 instances based on CPU utilization metrics.
* Document the configuration steps and simulate a load test to observe the scaling in action.