**AWS Must Do Solutions**

Section 1 : Introduction to AWS Services

Ques 2: Identify AWS Indian regions, Number of Availability Zones in that region, Edge Locations in India

Ans: AWS Indian Region ---> a. Asia Pacific (Mumbai) Region

b. Asia Pacific (Hyderabad) Region

Availability Zone ---> a. ap-south-1a d. ap-south-2a

b. ap-south-1b e. ap-south-2b

c. ap-south-1c f. ap-south-2c

Edge Locations: Mumbai, Delhi, Chennai, Bengaluru, Kolkata, Hyderabad

Section 2: IAM

q.1: Create IAM user attach user to a group with create a policy with just s3 read access. Attach the policy to the group

Ans: Step1: Go to IAM in services and select User-group from Dashboard from left panel

Step2: Go to user group in which u want to add policy

Step3: Go to permissions section - > Add Permissions - > Create Inline Policy

Step4: Choose service as s3, select Read as Actions allowed, select resources as all

Step5: Click next and add give the name of the policy and click create policy

q.2: Create a role which has read access to ec2 resources. Using aws cli assume the

role and list out the ec2 instances

Ans: Using command prompt we can make a role using the code below

step1: Create an IAM role with a trust policy

Code -> aws iam create-role --role-name EC2ReadOnlyRole --assume-role-policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Effect\": \"Allow\", \"Principal\": { \"AWS\": \"<Your-Account-ARN>\" }, \"Action\": \"sts:AssumeRole\" } ] }"

# attach the account arn in the code

Step2: Attach a policy to give read-only access to EC2 resources

Code: aws iam attach-role-policy --role-name EC2ReadOnlyRole --policy-arn arn:aws:iam::aws:policy/AmazonEC2ReadOnlyAccess

Step3: Assume the role and get temporary credentials  
Code: aws sts assume-role --role-arn arn:aws:iam::<account-id>:role/EC2ReadOnlyRole --role-session-name EC2ReadOnlySession

#Add ur account id here

Step4: Use the temporary credentials to list EC2 instances

Code: aws ec2 describe-instances --query "Reservations[\*].Instances[\*].Tags[?Key=='Name'].Value[]" --output text

This is used to list all the instances given in EC2.

Q3,

q.4) Create instance IAM profile and attach to instance

#### Ans: **1. Create the IAM Role**

Step1: Go to Roles in IAM Dashboard

Step2: Create Role > AWS Services > Permissions  
step3: Give a suitable name and create the role

#### **2. Create an Instance Profile for the Role**

Step1: Create a EC2 instance

An **Instance Profile** is required to associate the IAM role with an EC2 instance.

#### **3. Attach the IAM Role to an EC2 Instance**

Step1: Go to the Instance  
step2: Action > security > manage iam role

Step3: Now add on the role made by you

**SOME DEFINATIONS**

### **Instance Profiles**

An **Instance Profile** is a container for an IAM role that you can use to pass role information to an EC2 instance.

#### **Key Points:**

* It allows EC2 instances to assume an IAM role to access AWS services securely without embedding credentials in the instance.
* Each instance profile can contain only one role.
* Instance profiles are automatically created when you create a role for EC2 in the AWS Management Console.

### **STS (Security Token Service)**

AWS **Security Token Service (STS)** is a web service that allows you to request temporary, limited-privilege credentials.

#### **Key Features:**

* Temporary credentials are tied to a specific duration (up to 12 hours).
* Commonly used for cross-account access, federated access, or granting short-term access to applications.

### **Assume Roles**

The **Assume Role** feature allows you to grant permissions to entities (users, applications, or other roles) to temporarily take on the permissions of a different role.

#### **Key Points:**

* The entity assumes the role and receives temporary credentials from STS.
* Commonly used for cross-account access or delegating access within the same account.

**SECTION 3 : S3**

Q1. Create Bucket and put Objects

Ans: step1: Go to aws s3 services  
 step2: Click create S3 bucket  
 step3: Add features according to your choice

Step4: Click create bucket and you can now access the bucket

Q2. Setup Versioning

Ans: Step1: Go to S3 bucket -> Go to properties -> Enable Bucket Versioning

Step2: Now can see versions in bucket

Q3: Access Policies

Ans: Step1: Go to Bucket -> Go to permissions -> Go to Bucket policy

Step2: Edit the Policy -> Now click policy generator and add the details

Step3: After adding details you will get a policy code  
 Step4: Copy the code and add it to your policy  
 Step5: Now you can enjoy the policy features

Q4: Host a website from s3

Ans: Step1: Add the file of the html to get the website

Step2: Go to the properties section and go to static website hosting

Step3: Enable the static website hosting

Step4: Add the name of the html file there which you uploaded in bucket

Step5: Now save the changes and click the url you get below the static webiste hosting

Q5: Setup CRR

Ans: Step1: Create 2 instances

One where you will upload the files and second where files from first instance will replicate.

Step2: Upload the files in first instance

Step3: In first Instance - > Go to management - > Go to Replications Rule - > go to add new rule

Step4: Add the name of rule and destination instance which is our second instance

Step5: Now after saving changes, you can see the files in your replica instance

**Section 4: CLI and BOTO3**

Q1. **Using AWS CLI**

a. Create a s3 bucket with your name

b. Enable s3 versioning

c. Upload multiple versions of same file

d. List out all the versions of the file

Ans: Step1: Make bucket into command prompt

Command - > **aws s3 mb s3://<bucket\_name>**

Step2: Check the version with help of the below command

Command -> **aws s3api get-bucket-versioning –bucket**   **<bucket\_name>**

Step3:A. uploading a file called example.txt with text version 1

Command: **echo “version 1” > example.txt**

B. Upload it into ur bucket

Command: **aws s3 cp example.txt s3://<bucket\_name>/**

C. Modify the file now

Command: **echo “version 2” >> example.txt**

D. Reupload it

Command: **aws s3 cp exampe.txt s3://bucket\_name/**

This creates multiple vesions

Step4: List all the versions out here

Command: **aws s3api list-object-versions –bucket**  **<bucket\_name> --prefix example.txt**

**Q2. Write a python script use boto3 library which create a s3 bucket with transfer acceleration enabled**

Ans: Step1: Open Cmd Prompt , go to aws configure and add access key

Step2: Install boto3 : **Pip install boto3**

Step3: Open vscode and write the code for creating a s3 bucket and enable transfer acceleration

Step4: Go to cmd prompt and go to directory where u saved the file with help of cd - > **cd/Users/user/downloads**

Step5: Run it using - > **python main.py**

Step6: Check whether transer acceleration is enabled or not

Command: **aws s3api get-bucket-accelerate –configuration –** **bucket <bucket\_name>**

**Section 5: EC2**

**1. Launch a EC2 with both the OS flavour’s i.e linux and windows**

Ans: Simple step to launch a instance with linux ami

And using same ami make a instance with windows os

**2. Understand difference between private and public ip’s**  
Ans: Udemy Lecture

**3. Explore security group and Cidrs**

Ans : Creating new security groups and adding new connection to it

Cloudguru and Udemy Lectures

**4. Using Amazon Linux V2, t2.small Instance, Install nginx and create ami**

Ans: Step1: Create a EC2 instance using linux,t2.small ,allow http for nginx

Step2: Install Nginx with following commands

A. Open cmd prompt and connect Instance

Command: **ssh –i <key\_name>.pem ec2-user@ip\_address**

B. Update the system

Command: **sudo yum update –y**

C. Install nginx using command

Command: 1. **sudo yum install nginx –y**

2. **Sudo systemctl start nginx**

3. **Sudo systemctl enable nginx**

C. Open Chrome and type : [**http://ip\_address**](http://ip_address) and you will get nginx welcome page

**SECTION 6: CloudWatch**

Q1. **Create a Log Group**

Ans: Go to Aws - > Cloudwatch - > Log groups - > Create Log Groups

Q2. **Write a python script which writes logs to log groups (use boto3)**

Ans: Step1: Create a log group and give a name

Step2: Create a log stream inside it with proper name

Step3: Open a Vscode and write code to create log messages

Step4: Open command prompt -> aws configure -> python main.py

Step5:Check into log streams , the messages will be displayed into the log streams

Q3. **Create Cloudwatch monitoring Dashboard which provides instance cpu metrics**

Ans: 1. Go to Aws -> Cloudwatch -> Dashboard -> Create Dashboard -> Metrices

2. Below into browse section click EC2 -> Pre-instance Metrices ->

CpuUtilization

3. Now you will get a dashboard for cpu usage of the instance