IAM: Practical Scenarios & Governance

You get a "Access Denied" error while trying to access an S3 bucket from an EC2 instance. How would you troubleshoot?

🔍 1. Check EC2 IAM Role Association

- **Does your EC2 instance have an IAM role attached?**
  - Go to **EC2 Console > Instance > Description > IAM Role**.
- If **no role is attached**, the instance won't have permissions to access AWS services.
  - ✅ **Fix**: Attach an IAM role with appropriate permissions.

🔐 2. Check IAM Role Policies

- Go to **IAM Console > Roles > [Your EC2 Role] > Permissions**.
- Verify that the **IAM policy** allows access to the S3 bucket and actions like `s3:GetObject`, `s3:ListBucket`, etc.

✅ **Sample IAM policy**:

```
json
-----------
{
  "Effect": "Allow",
  "Action": ["s3:GetObject", "s3:ListBucket"],
  "Resource": [
    "arn:aws:s3:::your-bucket-name",
    "arn:aws:s3:::your-bucket-name/*"
  ]
}
```

🪪 3. Check S3 Bucket Policy

- Go to **S3 Console > Your Bucket > Permissions > Bucket Policy**.
- Ensure that the bucket policy does not **explicitly deny access** or is **not limited to a specific principal or condition**.

✅ **Example of good practice** (allow access from a specific role):

```
json
-----------
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::123456789012:role/YourEC2Role"
  },
  "Action": "s3:*",
  "Resource": [
    "arn:aws:s3:::your-bucket-name",
    "arn:aws:s3:::your-bucket-name/*"
  ]
}
```

🔒 4. Check for Bucket Ownership and ACLs

- If the S3 bucket is owned by another AWS account:
  - Make sure **bucket ACLs** or **bucket policy** allow your account or role access.
- Check **Object ACLs** as well (especially for `s3:GetObject`).

🌐 5. Check Network and Endpoint Configuration

- Is your EC2 instance in a **private subnet**?
  - If yes, and accessing S3 over the internet, you need:
    - **NAT Gateway** or
    - **VPC Endpoint for S3**
- If using **VPC Endpoint**, make sure:
  - Endpoint policy allows access.
  - Route tables and security groups permit traffic.

🧪 6. Test with AWS CLI on EC2

SSH into the EC2 instance and run:

```
bash
-----------
aws s3 ls s3://your-bucket-name --region your-region
```

- If this fails with `Access Denied`, the issue is definitely IAM or S3 policy related.

🧾 7. Check CloudTrail Logs

- Go to **CloudTrail** and search for `S3 AccessDenied` events.
- You can see **who made the request**, what policy was evaluated, and **why it was denied**.

Summary Checklist

Check	What to Verify
✅ IAM Role	Role is attached to EC2
✅ IAM Policy	Allows correct `s3:*` actions
✅ S3 Bucket Policy	No `Deny`, includes correct role/account
✅ Network	Internet/NAT Gateway or VPC Endpoint
✅ CLI Test	Run `aws s3 ls` to confirm
✅ CloudTrail	See exact reason for denial

You need to give read-only access to a specific S3 bucket to a team. How do you design the IAM policy?

✅ 1. IAM Policy for Read-Only Access to One S3 Bucket

Replace `your-bucket-name` with your actual bucket name.

```
json
-----------
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowListBucket",
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::your-bucket-name"
    },
    {
      "Sid": "AllowReadObjects",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetObjectAcl"
      ],
```

```
      "Resource": "arn:aws:s3:::your-bucket-name/*"
    }
  ]
}
```

✅ 2. How to Apply This Policy

You can apply this policy to:

- An **IAM group** (if the team members are in a group)
- An **IAM role** (for apps or EC2 instances)
- Individual **IAM users**

How would you implement automated IAM policy deployment across multiple accounts?
Use AWS CloudFormation StackSets or Terraform with automation pipelines (CI/CD) across accounts.

How do you implement a permission guardrail system using SCPs?

Create **Service Control Policies** in AWS Organizations to restrict or deny unwanted permissions at the org or OU level.

How do you implement compliance reporting for IAM permissions?
Use AWS IAM Access Analyzer, AWS Config rules, and AWS CloudTrail logs for auditing and reporting.

How would you design an IAM strategy that complies with specific regulations (e.g., HIPAA, PCI DSS)?

Enforce least privilege, enable MFA, use encryption, enable logging, and apply audit controls as per compliance.

How can you use IAM to implement cost control measures?

Restrict permissions to costly services, use SCPs or IAM policies to limit resource creation, and tag resources for cost tracking.

Which service can be used to audit IAM permissions and detect overly permissive roles?

A) AWS CloudWatch

B) AWS Config

C) AWS IAM Access Analyzer

D) AWS Shield

Answer: C) AWS IAM Access Analyzer

To comply with regulations like HIPAA, an IAM strategy should include:

A) Use of MFA and least privilege access

B) Granting full admin rights to all users

C) Disabling logging for privacy

D) Avoid encryption

Answer: A) Use of MFA and least privilege access

How can IAM help control AWS costs?

A) By granting unlimited access to billing services

B) By restricting resource creation permissions

C) By disabling CloudTrail

D) By enabling anonymous access

Answer: B) By restricting resource creation permissions