Example 1: Public Read Access to Bucket

✅ Using Bucket Policy (Recommended)

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "PublicReadGetObject",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::my-bucket-name/*"
  }]
}
```

🔹 This allows **anyone** on the internet to read (GET) objects in `my-bucket-name`.

Using ACL (Not Recommended for Public Access)

Set object-level ACL to public:

OR

```
aws s3api put-object-acl \
  --bucket my-bucket-name \
  --key example.txt \
  --acl public-read
```

🔹 This makes only `example.txt` publicly readable.
🔸 Risk: It's easy to forget or misconfigure object-level ACLs.

Example 2: Grant Access to Another AWS Account

✅ Using Bucket Policy

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "GrantCrossAccountAccess",
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::my-bucket-name/*"
  }]
}
```

🔹 Grants read access to **another AWS account** (`123456789012`).

Using ACL

```
aws s3api put-object-acl \
  --bucket my-bucket-name \
  --key example.txt \
  --grant-read id=canonical-user-id-of-other-account
```

🔸 You must know the **canonical user ID**, which is hard to find.
🔹 Works only at the object level — not efficient for entire buckets.

Example 3: Restrict Access by IP Address

✅ Using Bucket Policy

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "IPAllow",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": ["arn:aws:s3:::my-bucket-name",
"arn:aws:s3:::my-bucket-name/*"],
    "Condition": {
      "NotIpAddress": { "aws:SourceIp": "203.0.113.0/24" }
    }
  }]
}
```

🔹 Denies all actions **unless** the request comes from the IP range `203.0.113.0/24`.

❌ **ACLs can't do this** — only bucket policies support IP restrictions.

Example 4: Allow Only Encrypted Uploads (SSE-S3)

✅ Using Bucket Policy

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "DenyUnencryptedUploads",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::my-secure-bucket/*",
    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-server-side-encryption": "AES256"
      }
    }
  }]
}
```

🔒 Ensures all uploaded objects are encrypted using SSE-S3 (AES256).

❌ **ACLs cannot enforce encryption requirements.**

Example 5: Grant Temporary Access to a Specific IAM Role

✅ Using Bucket Policy

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowSpecificRoleAccess",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:role/TemporaryUploaderRole"
    },
    "Action": ["s3:PutObject"],
    "Resource": "arn:aws:s3:::my-bucket-name/uploads/*"
  }]
}
```

🎯 Grants the role permission to upload to a specific prefix.

❌ **ACLs cannot reference IAM roles**, only AWS account IDs or canonical user IDs.

Example 6: Make a Single File Public Using ACL

⚠️ Using ACL (only when needed)

```
aws s3api put-object-acl \
  --bucket my-bucket-name \
  --key public-info.txt \
  --acl public-read
```

🔸 This makes **only** **`public-info.txt`** publicly accessible.

🟡 **Use sparingly**; better to use bucket policy for more control.

Example 7: Grant Full Access to Another Account (ACL)

⚠️ Using ACL

```
aws s3api put-bucket-acl \
  --bucket my-bucket-name \
  --grant-full-control id=other-account-canonical-id
```

🔸 Grants full control to another AWS account using their **canonical ID**.

🛑 **Cons**:

- Canonical IDs are hard to manage.
- No way to restrict by IP, time, or object prefix.

Example 8: Block All Public Access (Recommended)

✅ Using Bucket Settings (not policy or ACL directly)

This is done via the **S3 Block Public Access settings** (in console or CLI):

```
aws s3api put-public-access-block \
  --bucket my-bucket-name \
  --public-access-block-configuration \
    BlockPublicAcls=true \
    IgnorePublicAcls=true \
    BlockPublicPolicy=true \
    RestrictPublicBuckets=true
```

🔒 Prevents **any public access**, even if ACL or bucket policy tries to allow it.