MinIO Policies

MinIO’s access policies are managed through JSON-based policy definitions. These policies govern access control, specifying which actions are allowed or denied on resources (buckets, objects, etc.) based on certain conditions. You can define granular access permissions for users or groups using these JSON policies.

JSON Policy Structure

A MinIO JSON policy generally has the following structure:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow" | "Deny",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        ...
      ],
      "Resource": [
        "arn:aws:s3:::bucket-name/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

Example 1: Multiple Statements with Mixed Permissions

In this example, a policy grants full access to one bucket and read-only access to another bucket

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::first-bucket",
        "arn:aws:s3:::first-bucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::second-bucket",
        "arn:aws:s3:::second-bucket/*"
      ]
    }
  ]
}
```

Explanation:

- **First Statement**: Grants full access (list, upload, download, delete) to the bucket `first-bucket`.
- **Second Statement**: Grants read-only access (list and get) to the bucket `second-bucket`

Example 2: Mixed Permissions with IP Condition

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::restricted-bucket",
        "arn:aws:s3:::restricted-bucket/*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "192.168.1.0/24"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::restricted-bucket",
        "arn:aws:s3:::restricted-bucket/*"
      ],
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "192.168.1.0/24"
        }
      }
    }
  ]
}
```

Explanation:

- **First Statement**: Allows full access to `restricted-bucket` only from the IP range `192.168.1.0/24`.
- **Second Statement**: Denies all actions on the same bucket if the request comes from any IP address outside of the specified range.

Example 3: Object Tagging and Metadata Management

```
This policy allows a user to manage object metadata (e.g., adding or
removing tags), but not the actual object contents.
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObjectTagging",
        "s3:GetObjectTagging",
        "s3:DeleteObjectTagging",
        "s3:PutObjectAcl",
        "s3:GetObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::metadata-bucket/*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::metadata-bucket/*"
      ]
    }
  ]
}
```

Explanation:

- **First Statement**: Allows the user to manage tags and ACLs (Access Control Lists) on objects in the `metadata-bucket`.
- **Second Statement**: Denies the user permission to modify or access the actual object data itself.

Example 4: Granting Access to Specific Prefixes in a Bucket

In this example, the policy allows read and write access only to objects that begin with a certain prefix in a bucket.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::example-bucket/docs/*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "s3:PutObject",
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::example-bucket/*"
      ],
      "Condition": {
        "StringNotLike": {
          "s3:prefix": "docs/*"
        }
      }
    }
  ]
}
```

Explanation:

- **First Statement**: Grants full access to the objects under the prefix `docs/` in the `example-bucket`.
- **Second Statement**: Denies access to any objects in the bucket that do not have the prefix `docs/`.

Example 5: Combining Bucket and Object Level Permissions

This example shows a policy that allows listing the objects in the bucket and full control over objects, but does not allow bucket deletion.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket/*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "s3:DeleteBucket"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket"
      ]
    }
  ]
}
```

Explanation:

- **First Statement**: Allows listing of the bucket (`my-bucket`).
- **Second Statement**: Grants full control over the objects within the bucket (get, put, delete).
- **Third Statement**: Explicitly denies deletion of the bucket itself.

Example 6: Expiry Policy with Lifecycle Action

In this example, a condition is added for specific access to objects only within a certain time frame.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::time-sensitive-bucket/*"
      ],
      "Condition": {
        "DateLessThan": {
          "aws:CurrentTime": "2024-12-31T23:59:59Z"
        }
      }
    }
  ]
}
```

Explanation:

- **Single Statement**: Allows retrieving objects from the `time-sensitive-bucket` only until the specified date (`2024-12-31T23:59:59Z`). After this date, access is automatically denied.

Example 7: Restrict Access Based on Time of Day

This policy allows read access to a bucket, but only between 9 AM and 5 PM UTC.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::time-restricted-bucket/*"
      ],
      "Condition": {
        "DateGreaterThan": {
          "aws:CurrentTime": "2024-09-05T09:00:00Z"
        },
        "DateLessThan": {
          "aws:CurrentTime": "2024-09-05T17:00:00Z"
        }
      }
    }
  ]
}
```

Example 8: Restrict Access Based on User-Agent (Client Application)

This policy allows uploading objects, but only if the request comes from a specific user-agent (for example, a trusted client application).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::app-specific-bucket/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:UserAgent": "trusted-app/1.0"
        }
      }
    }
  ]
}
```

Explanation:

- **Single Statement**: Allows uploading objects (`s3:PutObject`) to `app-specific-bucket` only if the request is made using the user-agent string `trusted-app/1.0`. This is useful for controlling access from specific applications.

Example 9: IP Address Whitelisting and Blacklisting

This policy allows access to objects from a specified IP range (whitelisting) but explicitly denies access from a different range (blacklisting).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::whitelisted-bucket/*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "192.168.0.0/24"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "s3:*"
      ],
      "Resource": [
        "arn:aws:s3:::whitelisted-bucket/*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "10.0.0.0/24"
        }
      }
    }
  ]
}
```

Explanation:

- **First Statement**: Allows access to all objects in `whitelisted-bucket` from IP addresses in the range `192.168.0.0/24`.
- **Second Statement**: Explicitly denies access to all objects in the same bucket from IP addresses in the range `10.0.0.0/24`.

Example 10: Restrict Access Based on SecureTransport (HTTPS Enforcement)

This policy allows read/write access to objects but only if the request is made using HTTPS (secure transport).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::secure-bucket/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

Explanation:

- **Single Statement**: Grants access to `secure-bucket` only when the request is made over HTTPS (`aws:SecureTransport` is `true`). This ensures that data is transferred securely.

Example 11: Allow Access Based on Object Size (Using Object Tags)

This policy allows access to objects that have a specific object tag indicating their size (for example, files smaller than 1 MB).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::size-restricted-bucket/*"
      ],
      "Condition": {
        "NumericLessThanEquals": {
          "s3:object-size": 1048576
        }
      }
    }
  ]
}
```

Explanation:

- **Single Statement**: Grants access to objects in `size-restricted-bucket` only if their size is 1 MB or smaller (`1048576` bytes). You can use this for enforcing access control based on object size

Example 12: Restrict Access Based on File Extension

This policy restricts access based on the file extension. It only allows access to objects with `.txt` extension.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::extension-restricted-bucket/*.txt"
      ]
    }
  ]
}
```