

# Deploying Infrastructure with Terraform:

## Authentication and Authorization:

### Authentication:

Authentication verifies the identity of users, services, or systems accessing your AWS infrastructure.

### AWS Authentication Methods:

1. **Access Keys/secret key:** Use access keys to authenticate API requests.
2. **IAM Roles:** Use IAM roles to delegate access to AWS resources.
3. **AWS STS:** Use AWS Security Token Service (STS) to request temporary credentials.

### Terraform AWS Authentication Examples

#### 1. Access Keys:

```
Terraform > expense > Authentication& Authorization.tf
1
2  provider "aws" {
3    region    = "us-west-2"
4    access_key = "YOUR_ACCESS_KEY"
5    secret_key = "YOUR_SECRET_KEY"
6  }
7
```

#### 2. \*IAM Roles\*:

```
10
11  provider "aws" {
12    region = "us-west-2"
13    assume_role {
14      role_arn = "arn:aws:iam::123456789012:role/example-role"
15    }
16  }
17
```

## 1. AWS STS:

```
18
19 provider "aws" {
20   region = "us-west-2"
21   assume_role {
22     role_arn = "arn:aws:iam::123456789012:role/example-role"
23     external_id = "YOUR_EXTERNAL_ID"
24   }
25 }
26
```

### \*Authorization\*

Authorization determines what actions authenticated users, services, or systems can perform on your AWS infrastructure.

### \*AWS Authorization Methods\*

1. **\*IAM Policies\***: Attach policies to users, groups, or roles.
2. **\*Resource-Based Policies\***: Attach policies directly to resources.
3. **\*Permissions Boundaries\***: Define permissions boundaries for IAM entities.

### \*Terraform AWS Authorization Examples\*

#### 1. \*IAM Policy\*:

```
28
29 resource "aws_iam_policy" "example" {
30   name           = "example-policy"
31   description    = "Example policy"
32
33   policy         = jsonencode({
34     Version = "2012-10-17"
35     Statement = [
36       {
37         Action = [
38           "ec2:DescribeInstances",
39           "s3:ListBucket"
40         ]
41         Resource = "*"
42         Effect   = "Allow"
43       }
44     ]
45   })
46 }
47
```

## 2. Resource-Based Policy:

```
48
49 resource "aws_s3_bucket_policy" "example" {
50   bucket = (link unavailable)
51
52   policy = jsonencode({
53     Version = "2012-10-17"
54     Statement = [
55       {
56         Action = [
57           "s3:GetObject"
58         ]
59         Resource = "${aws_s3_bucket.example.arn}/*"
60         Effect   = "Allow"
61       }
62     ]
63   })
64 }
65
```

## 3. \*Permissions Boundary\*:

```
68 resource "aws_iam_permissions_boundary" "example" {
69   permissions_boundary_type = "Organization"
70   permissions_boundary_arn = aws_iam_policy.example.arn
71 }
72
73
```

By implementing authentication and authorization, you ensure secure access to your AWS infrastructure and control what actions can be performed.

## Real-time Example:

Suppose we have a Terraform configuration that provisions an AWS EC2 instance and an S3 bucket. We want to authenticate using AWS IAM and authorize access to the instance and bucket.

```

71 # Authenticate using AWS IAM
72 ✓ provider "aws" {
73     region = "us-west-2"
74     ✓ assume_role {
75         role_arn = "arn:aws:iam::123456789012:role/example-role"
76     }
77 }
78
79 # Create EC2 instance
80 ✓ resource "aws_instance" "example" {
81     ami          = "ami-abc123"
82     instance_type = "t2.micro"
83 }
84
85 # Create S3 bucket
86 ✓ resource "aws_s3_bucket" "example" {
87     bucket = "example-bucket"
88 }
89

```

```

89
90 # Create IAM policy for EC2 instance and S3 bucket
91 resource "aws_iam_policy" "example" {
92     name          = "example-policy"
93     description   = "Example policy"
94
95     policy        = jsonencode({
96         Version = "2012-10-17"
97         Statement = [
98             {
99                 Action = [
100                     "ec2:DescribeInstances",
101                     "s3:ListBucket"
102                 ]
103                 Resource = "*"
104                 Effect    = "Allow"
105             }
106         ]
107     })
108 }
109
110 # Attach IAM policy to EC2 instance and S3 bucket
111 resource "aws_iam_role_policy_attachment" "example" {
112     role          = aws_iam_role.example.name
113     policy_arn    = aws_iam_policy.example.arn
114 }
115

```

This example demonstrates authentication using AWS IAM and authorization using IAM policies and roles.