Terraform S3 with IAM Admin Access - Code Explanation

Terraform Script with Line-by-Line Explanation

```
resource "aws_s3_bucket" "my_bucket1" {  # Creates an S3 bucket named 'ansible.co.in'
  bucket = "ansible.co.in"               # Unique bucket name
  tags = {
    Name = "pract"                       # Adds a tag to the bucket
  }
}

resource "aws_s3_object" "terraform_state" {  # Uploads a file to the S3 bucket
  bucket = aws_s3_bucket.my_bucket1.id        # Reference to the S3 bucket
  key    = "terraform.tfstate.backup"         # The object name (key) in S3
   source = "C:/Users/Acer/Desktop/terraform/s3bucket/terraform.tfstate.backup"  # Local file to
upload
  acl    = "private"                          # Access control list (private access)
}

resource "aws_s3_object" "cover_letter" {
  bucket = aws_s3_bucket.my_bucket1.id
  key    = "coverletter.txt"
  source = "C:/Users/Acer/Desktop/terraform/s3bucket/covelatter.txt"
  acl    = "private"
}

resource "aws_iam_policy" "s3_admin_policy" {  # Defines a custom IAM policy
  name        = "S3BucketAdminPolicy"          # Name of the policy
  description = "Full admin access to the ansible.co.in bucket"

  policy = jsonencode({                        # Encodes JSON policy
    Version = "2012-10-17",                    # IAM policy language version
    Statement = [{
      Effect = "Allow",                        # Allows the action
      Action = "s3:*",                         # Grants all S3 permissions
      Resource = [
        aws_s3_bucket.my_bucket1.arn,          # Bucket ARN
        "${aws_s3_bucket.my_bucket1.arn}/*"    # All objects within the bucket
      ]
    }]
  })
}

resource "aws_iam_user" "admin_user" {         # Creates a new IAM user
  name = "admin-user"
}

resource "aws_iam_user_policy_attachment" "admin_attach" {  # Attaches the policy to the user
  user       = aws_iam_user.admin_user.name      # User name
  policy_arn = aws_iam_policy.s3_admin_policy.arn  # ARN of the policy created
```

```
}
```