```
resource "aws_iam_role" "lambda_exec_role" {
  name = "lambda_execution_role"

  assume_role_policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Action = "sts:AssumeRole"
        Principal = {
          Service = "lambda.amazonaws.com"
        }
        Effect = "Allow"
        Sid = ""
      },
    ]
  })
}

resource "aws_iam_role" "ec2_s3_access_role" {
  name = "ec2_s3_access_role"

  assume_role_policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      {
        Effect = "Allow",
```

```
        Principal = {
          Service = "ec2.amazonaws.com",
        },
        Action = "sts:AssumeRole",
      },
    ],
  })
}

resource "aws_iam_policy" "ec2_s3_access" {
  name        = "ec2_s3_access_policy"
  description = "Allow EC2 instances to access the S3 bucket"

  policy = jsonencode({
    Version = "2012-10-17",
    Statement = [
      {
        Action = [
          "s3:GetObject",
          "s3:ListBucket",
              "s3:PutObject" # Include PutObject if you need write access, for example, to upload the kubeconfig file.
        ],
        Resource = [
          "arn:aws:s3:::swarmskube",
          "arn:aws:s3:::swarmskube/*"
```

```
        ],
        Effect = "Allow",
      },
    ],
  })
}

resource "aws_iam_instance_profile" "ec2_instance_profile" {
  name = "ec2_instance_profile"
  role = aws_iam_role.ec2_s3_access_role.name
}

resource "aws_iam_policy_attachment" "ec2_s3_access_attachment" {
  name       = "ec2_s3_access_attachment" # Correct argument for specifying the attachment name
  policy_arn = aws_iam_policy.ec2_s3_access.arn
  roles      = [aws_iam_role.ec2_s3_access_role.name]
}
```