IAM Implementation Guide

Prerequisites

1. AWS account with appropriate permissions
2. AWS CLI installed and configured
3. Basic understanding of JSON

Step 1: Creating IAM Users

```
# Create IAM user
aws iam create-user --user-name JohnDoe

# Create access key for the user
aws iam create-access-key --user-name JohnDoe

# Create login profile for console access
aws iam create-login-profile --user-name JohnDoe --password
'TempPassword123!' —password-reset-required
```

Step 2: Creating IAM Groups

```
# Create group
aws iam create-group --group-name Developers

# Add user to group
aws iam add-user-to-group --user-name JohnDoe --group-name Developers
```

Step 3: Creating IAM Policies

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

```
# Create policy
aws iam create-policy --policy-name S3DevAccess --policy-document file://policy.json

# Attach policy to group
aws iam attach-group-policy --group-name Developers --policy-arn
arn:aws:iam::123456789012:policy/S3DevAccess
```

Step 4: Creating IAM Roles

```
# Create trust policy for EC2
cat > trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF

# Create role
aws iam create-role --role-name EC2S3AccessRole --assume-role-policy-document file://trust-
policy.json

# Attach policy to role
aws iam attach-role-policy --role-name EC2S3AccessRole --policy-arn arn:aws:iam::aws:policy/
AmazonS3ReadOnlyAccess
```

Step 5: Setting Up MFA

```
# Enable MFA for user
aws iam create-virtual-mfa-device --virtual-mfa-device-name JohnDoeMFA --outfile
base32string.txt --bootstrap-method QRCodePNG

# Associate MFA device with user
aws iam enable-mfa-device --user-name JohnDoe --serial-number
arn:aws:iam::123456789012:mfa/JohnDoeMFA --authentication-code1 123456 --
authentication-code2 789012
```

Best Practices

1. Least Privilege Access

- Grant only the permissions required to perform tasks
- Start with no permissions and add as needed
- Use IAM Access Analyzer to review policies
- Regularly audit permissions with IAM reports

2. Use Strong Authentication

- Enable MFA for all IAM users
- Rotate access keys regularly
- Use temporary credentials instead of long-term keys
- Implement password policies

3. Centralize Permission Management

- Use groups to assign permissions to users
- Use roles for applications and AWS services
- Implement permissions boundaries
- Use AWS Organizations for multi-account management

4. Monitor and Audit

- Enable CloudTrail across all regions
- Set up CloudWatch Alarms for suspicious activity
- Regularly review IAM access reports
- Use AWS Config to track IAM configuration change

5. Secure Root Account

- Don't use root account for daily tasks
- Enable MFA for root account
- Create strong root password
- Delete root access keys