Complete IAM Role Creation Steps for GitHub Actions

Prerequisites: Create OIDC Identity Provider (One-Time Setup)

Step 1: Navigate to Identity Providers

1. Open AWS Console
2. Go to **IAM** service
3. In left sidebar, click **Identity providers**
4. Click **Add provider**

Step 2: Configure OIDC Provider

1. **Provider type**: Select **OpenID Connect**
2. **Provider URL**: Enter `https://token.actions.githubusercontent.com`
3. Click **Get thumbprint** (AWS will auto-populate)
4. **Audience**: Enter `sts.amazonaws.com`
5. Click **Add provider**

Step 3: Verify Provider Creation

- You should see the provider listed as:
  - **Provider**: `token.actions.githubusercontent.com`
  - **Type**: `OpenID Connect`
  - **Audiences**: `sts.amazonaws.com`

Main Process: Create IAM Role

Step 1: Start Role Creation

1. In IAM, go to **Roles** in left sidebar
2. Click **Create role**

Step 2: Select Trusted Entity Type

1. **Trusted entity type**: Select **Web identity** ⭐ (This is key!)
2. **Identity provider**: Select `token.actions.githubusercontent.com` (the provider you created)
3. **Audience**: Select `sts.amazonaws.com`

Step 3: Configure GitHub Details

1. **GitHub organization**: Enter your GitHub username (e.g., `johnsmith`)
2. **GitHub repository**: Enter your repository name (e.g., `aws-s3-demo`)
3. **GitHub branch**: Leave empty (allows all branches) or specify `main`
4. Click **Next**

Step 4: Add Permissions (Skip AWS Managed Policies)

1. **DO NOT** select any AWS managed policies yet
2. Just click **Next** (we'll add custom policy later)

Step 5: Name and Review

1. **Role name**: `GitHubActions-S3Deploy-Role`
2. **Description**: `Role for GitHub Actions to deploy static websites to S3`
3. **Trusted entities**: Verify it shows `token.actions.githubusercontent.com`
4. Click **Create role**

Add Custom Permission Policy

Step 1: Open the Created Role

1. Go to **IAM** → **Roles**
2. Find and click `GitHubActions-S3Deploy-Role`

Step 2: Add Inline Policy

1. Click **Add permissions** dropdown
2. Select **Create inline policy**
3. Click **JSON** tab

Step 3: Add S3 Permissions Policy

Replace the entire JSON with:

```
json
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::YOUR-BUCKET-NAME",
        "arn:aws:s3:::YOUR-BUCKET-NAME/*"
      ]
    }
  ]
}
```

⚠️ **Important**: Replace `YOUR-BUCKET-NAME` with your actual S3 bucket name

Step 4: Save Policy

1. Click **Next**
2. **Policy name**: `S3DeploymentPolicy`
3. Click **Create policy**

Verify Role Configuration

Final Role Summary Should Show:

Trust relationships:

```
json
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::YOUR-ACCOUNT-ID:oidc-provider/token.actions.githubusercontent.com"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "token.actions.githubusercontent.com:aud": "sts.amazonaws.com"
        },
        "StringLike": {
          "token.actions.githubusercontent.com:sub": "repo:YOUR-USERNAME/YOUR-REPO:*"
        }
      }
    }
  ]
}
```

Permissions:

- One inline policy: `S3DeploymentPolicy`

Role ARN:

- Copy this: `arn:aws:iam::YOUR-ACCOUNT-ID:role/GitHubActions-S3Deploy-Role`
- You'll need this for GitHub Secrets

Key Information to Note Down

For Your Notebook:

1. **Role ARN**: `arn:aws:iam::YOUR-ACCOUNT-ID:role/GitHubActions-S3Deploy-Role`
2. **S3 Bucket Name**: Your bucket name
3. **AWS Region**: Your bucket's region
4. **GitHub Repository**: Full path (username/repository)

For GitHub Secrets:

- `AWS_ROLE_ARN`: The role ARN from above
- `S3_BUCKET_NAME`: Your S3 bucket name

- `AWS_REGION`: Your AWS region (e.g., `us-east-1`)

Common Issues and Solutions

Issue 1: "Web identity" option not available

Solution: Create OIDC Identity Provider first (see prerequisites)

Issue 2: GitHub provider not in dropdown

Solution: Refresh page, or create OIDC provider manually using CLI

Issue 3: Role creation fails with trust policy error

Solution: Ensure GitHub username/repository are exactly correct

Issue 4: "Access denied" when GitHub Actions runs

Solutions:

- Verify S3 bucket name matches in policy
- Check repository name in trust policy
- Ensure OIDC provider audience is `sts.amazonaws.com`

Validation Steps

Test the Setup:

1. Role should appear in IAM Roles list
2. Trust policy should reference your specific GitHub repo
3. Permission policy should allow S3 actions on your bucket
4. Role ARN should be properly formatted

Ready for GitHub Actions:

- ✅ OIDC Provider created
- ✅ IAM Role created with Web identity trust
- ✅ Custom S3 permissions added
- ✅ Role ARN copied for GitHub Secrets
- ✅ All placeholder values replaced with actual values

What We Just Created (Technical Summary)

Identity Provider: Tells AWS to trust GitHub-issued OIDC tokens **Trust Policy**: Allows only your specific GitHub repository to assume the role **Permission Policy**: Grants S3 access to deploy your static website **Temporary Credentials**: GitHub Actions gets 1-hour credentials, no permanent keys stored

This setup ensures:

- Zero long-term credentials stored anywhere
- Repository-specific access (no other repos can use this role)
- Automatic credential rotation
- Full audit trail in CloudTrail