**Automated AWS Authentication with OIDC and GitHub Actions**

**Step 1:** Create an OIDC Identity Provider in AWS

Sign in to the AWS Management Console and open the IAM console at IAM Console.

In the navigation pane, choose Identity providers and then choose **Add provider.**

For Configure provider, choose OpenID Connect (OIDC).

Type a name for the identity provider (e.g., GitHubOIDCProvider).

For Provider URL, enter https://token.actions.githubusercontent.com.

For Audience, enter sts.amazonaws.com.

Choose Add provider.

**Step 2:** Create an IAM Role for the OIDC Provider

In the IAM console, choose Roles and then Create role.

Choose the type of trusted entity as Web identity.

Select the identity provider you created (e.g., GitHubOIDCProvider).

Select the audience (e.g., sts.amazonaws.com).

Choose Next: Permissions.

Attach the necessary policies to the role (e.g., AdministratorAccess for admin permissions).

Choose Next: Tags (optional) and then Next: Review.

Type a name for the role (e.g., GitHubOIDC).

Choose Create role.

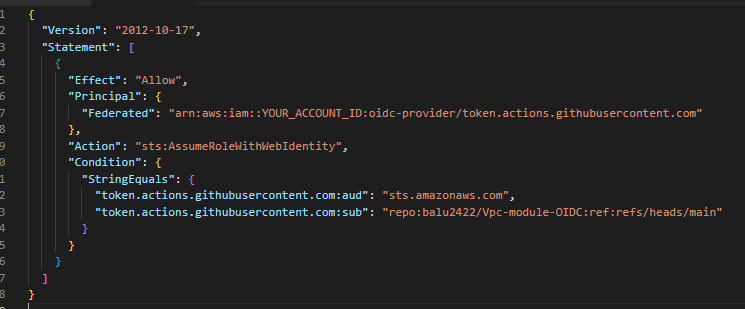
**Step 3:** Configure the Trust Policy for the Role

In the IAM console, select the role you created (e.g., GitHubOIDC).

Choose the Trust relationships tab and then Edit trust relationship.

Update the trust policy to allow GitHub Actions to assume the role

example trust policy:



**Explanation of the Trust Policy:**

Version: Specifies the version of the policy language.

Statement: Contains the policy statements.

Effect: Specifies whether the statement allows or denies access. Here, it is set to Allow.

Principal: Specifies the trusted entity. Here, it is the OIDC provider (token.actions.githubusercontent.com).

Action: Specifies the action that is allowed. Here, it is sts:AssumeRoleWithWebIdentity, which allows the role to be assumed using web identity.

Condition: Specifies the conditions under which the role can be assumed.

StringEquals: Ensures that the audience (aud) is sts.amazonaws.com and the subject (sub) matches the specific GitHub repository and branch (repo:balu2422/Vpc-module-OIDC:ref:refs/heads/main).

**Example GitHub Actions Workflow**

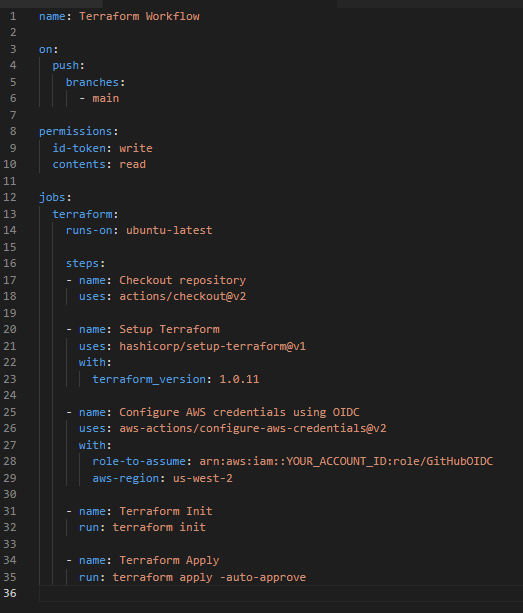
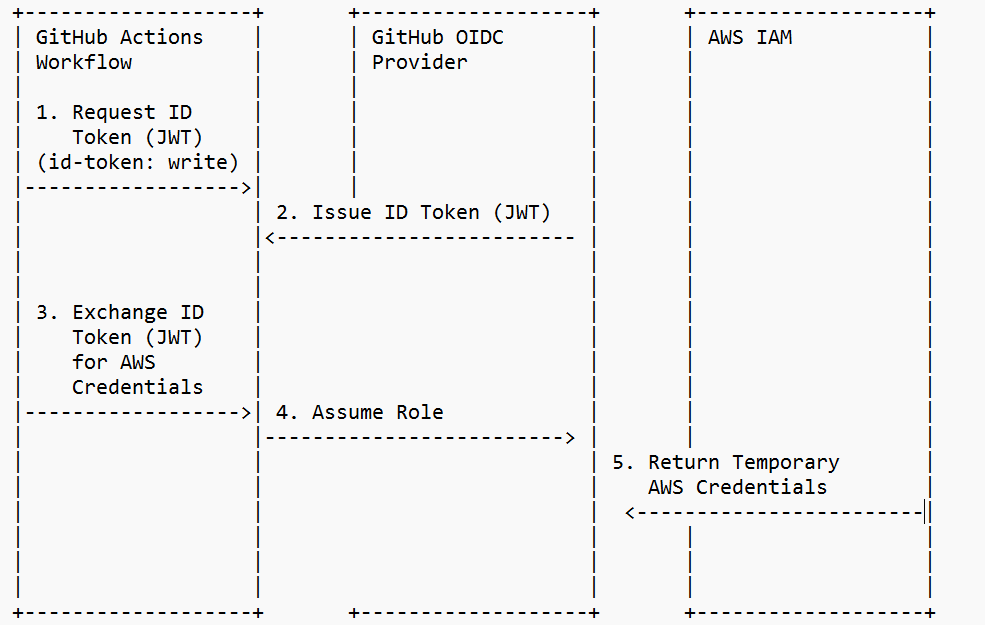


Diagram: OIDC Authentication Flow



GitHub Actions Workflow Requests ID Token (JWT):

**What:** The GitHub Actions workflow requests an ID token from GitHub's OIDC provider.

**Why:** The ID token is needed to authenticate the workflow with AWS.

**How:** The id-token: write permission allows the workflow to request an ID token. This permission sets two environment variables: ACTIONS\_ID\_TOKEN\_REQUEST\_URL and ACTIONS\_ID\_TOKEN\_REQUEST\_TOKEN, which are used to generate the ID token1.

**GitHub OIDC Provider Issues ID Token (JWT):**

**What**: GitHub's OIDC provider issues an ID token to the workflow.

**Why:** The ID token is a signed JWT (JSON Web Token) that proves the identity of the workflow.

**How:** The OIDC provider generates the token and sends it back to the workflow. This step starts because the workflow requested the ID token using the id-token: write permission.

**Workflow Exchanges ID Token (JWT) for AWS Credentials:**

**What:** The workflow exchanges the ID token for temporary AWS credentials.

**Why:** The temporary credentials are needed to perform actions in AWS.

**How:** The workflow uses the aws-actions/configure-aws-credentials action to assume an IAM role in AWS. This step starts because the workflow has received the ID token and needs to authenticate with AWS.

AWS IAM Assumes Role:

**What**: AWS IAM verifies the ID token and assumes the specified role.

**Why:** Assuming the role grants the workflow the necessary permissions to access AWS resources.

**How:** AWS IAM checks the trust policy of the role to ensure it trusts the GitHub OIDC provider and the specific repository. This step starts because AWS IAM received the ID token and verified its validity.

**AWS IAM Returns Temporary AWS Credentials:**

**What:** AWS IAM returns temporary AWS credentials to the workflow.

**Why:** The temporary credentials allow the workflow to authenticate and perform actions in AWS.

**How**: AWS IAM generates the credentials and sends them back to the workflow. This step starts because AWS IAM successfully assumed the role and generated the temporary credentials.