

**Step 1: Create the Web Application**

First, let's create a simple web application that displays "Hello, World!".

**index.html**

<!DOCTYPE html>

<html>

<head>

<title>Hello World App</title>

</head>

<body>

<h1>Hello, World!</h1>

</body>

</html>

**Step 2: Set Up AWS Cognito for SSO**

1. **Create a Cognito User Pool**:
   * Go to the AWS Cognito console.
   * Create a new user pool with the necessary settings.
   * Configure the app client settings and domain name.
2. **Configure App Client**:
   * Enable OAuth 2.0 settings for the app client.
   * Set up the callback URLs and sign-out URLs.

**Step 3: Deploy the Application Using Elastic Beanstalk**

1. **Create an Elastic Beanstalk Application**:
   * Go to the AWS Elastic Beanstalk console.
   * Create a new application and environment.
   * Upload the index.html file as the application source.

**Step 4: Use Terraform for Infrastructure as Code**

Create a main.tf file to define the infrastructure.

**main.tf**

provider "aws" {

region = "us-west-2"

}

resource "aws\_elastic\_beanstalk\_application" "hello\_world\_app" {

name = "hello-world-app"

description = "Hello World Application"

}

resource "aws\_elastic\_beanstalk\_environment" "hello\_world\_env" {

name = "hello-world-env"

application = aws\_elastic\_beanstalk\_application.hello\_world\_app.name

solution\_stack\_name = "64bit Amazon Linux 2 v3.1.2 running Node.js 14"

setting {

namespace = "aws:elasticbeanstalk:application:environment"

name = "NODE\_ENV"

value = "production"

}

}

**Step 5: Set Up CI/CD Pipeline with GitHub Actions**

Create a .github/workflows/deploy.yml file in your repository.

**deploy.yml**

name: Deploy to AWS Elastic Beanstalk

on:

push:

branches:

- main

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v2

- name: Set up Node.js

uses: actions/setup-node@v2

with:

node-version: '14'

- name: Install dependencies

run: npm install

- name: Deploy to Elastic Beanstalk

env:

AWS\_ACCESS\_KEY\_ID: ${{ secrets.AWS\_ACCESS\_KEY\_ID }}

AWS\_SECRET\_ACCESS\_KEY: ${{ secrets.AWS\_SECRET\_ACCESS\_KEY }}

AWS\_REGION: 'us-west-2'

run: |

zip -r deploy.zip .

aws elasticbeanstalk create-application-version --application-name hello-world-app --version-label v1 --source-bundle S3Bucket=my-bucket,S3Key=deploy.zip

aws elasticbeanstalk update-environment --environment-name hello-world-env --version-label v1

**Step 6: Documentation**

Create a README.md file in your repository.

**README.md**

# Hello World Application

## Architecture Diagram

!Architecture Diagram

## CI/CD Pipeline

1. \*\*Checkout Code\*\*: Pulls the latest code from the repository.

2. \*\*Set up Node.js\*\*: Configures the Node.js environment.

3. \*\*Install Dependencies\*\*: Installs necessary dependencies.

4. \*\*Deploy to Elastic Beanstalk\*\*: Deploys the application to AWS Elastic Beanstalk.

## Testing and Verification

1. \*\*Access the Application\*\*: Visit the URL provided by Elastic Beanstalk.

2. \*\*SSO Login\*\*: Ensure you are redirected to the Cognito login page.

3. \*\*Unauthorized Access\*\*: Verify unauthorized users cannot access the application.

## Submission

Please provide the link to your GitHub repository once completed.

**Final Steps**

1. **Push your code to GitHub**.
2. **Set up AWS credentials in GitHub Secrets**.
3. **Run the GitHub Actions workflow**.

Feel free to ask if you have any questions or need further assistance!