**Cloud Resume Challenge**

**Steps:**

**Step 1: Certification:**

I have already achieved AWS SAA and Cloud Practitioner certificates.

**Step 2: HTML Page**

Designed a html page containing my resume.

**Step 3: CSS**

My resume html page is also styled by CSS.

**Step 4:**  **Static website hosting**

Unchecked block public access

Updated bucket policy

Enabled Static website hosting

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

**Step 5: HTTPS**

The S3 website URL should use [HTTPS](https://www.cloudflare.com/learning/ssl/what-is-https/) for security. You will need to use [Amazon CloudFront](https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/) to help with this.

Create a CloudFront distribution with HTTPS only.

Then in S3-> checked block public access

Updated bucket policy from CF Dist

Disabled Static website hosting

Update default root object to index.html

A screenshot of a computer

Description automatically generated

**Step 6: DNS**

Request a public certificate in ACM.

Purchase a domain in Route53. (hemanthcloud.com)

**Step 7: Javascript**

Our resume template has js.

**Step 8: Database- DynamoDB**

Create a DynamoDB table – cloudresume-test

Create item-> id and views

A screenshot of a computer

Description automatically generated

**Step 9: API (through Lambda function)**

Create a lambda function-> cloud-resume-test-api

Lang-> latest python version

Adv setting🡪 Enable func URL

Auth type – none

Enable CORS

Write lambda func and enable fullDynamoDB access permissions

Create index.js and modify index.html

A screenshot of a computer

Description automatically generated

**Step 10: Python**

Created a lambda func using python n boto3.

**Step 11: Tests**

Included some tests for your Python code.

**Step 12: IAC**

**Step 13: Source Control-GIT**

**A screenshot of a computer

Description automatically generated**

**Step 14: CI/CD(Backend)**

**Step 15: CI/CD(Frontend)**

Setup a github repository

Add all previously created files into git.

Create front-end-CICD.yml file in .github/workflows and check actions.

Confige secrect access keys in settings

And rerun all jobs.[ jakejarvis/s3-sync-action🡪 Action on Github-marketplace]

A screenshot of a computer

Description automatically generated

**Step 16: Blog**

**IAC-Terraform**

Create main.tf, provider.tf, func.py files

Install terrrafrom

Configure aws n aws cli

$terraform init

$terraform plan

$terraform apply

$terraform destroy(to clean up resources)

**A screenshot of a computer

Description automatically generated** **A screenshot of a computer

Description automatically generated**