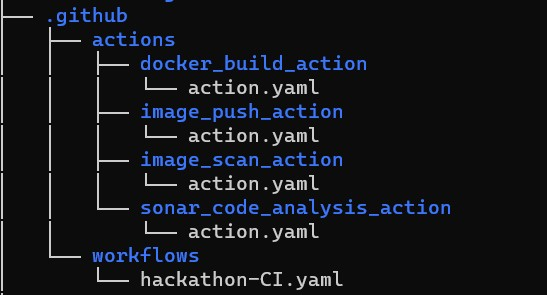
**Kaibur Assessment**

**Table of Contents**

* **Overview**
  + In this document, I have successfully implemented a Continuous Integration (CI) pipeline for a given Node.js application. This pipeline includes thorough code quality checks and image scanning. Additionally, I have implemented Continuous Deployment (CD) using Helm charts. Furthermore, I have seamlessly configured an Azure Virtual Machine (VM) using Terraform and accomplished the installation of MongoDB utilizing Ansible.
* **Prerequisites**
  + GitHub account, Virtual machine, Any one cloud for Testing, Kubernetes, Helm, Terraform, Ansible, Argo CD
* **GitHub Repository Structure**
  + Top of the repo inside**. GitHub** folder I have created action and workflow files for **CI and CD** workflow, I kept the helm charts inside the Helm folder and Terraform files with the name of Terraform and you can find the ansible also.
* **GitHub Actions Workflow**
  + Below is the Action Workflow structure:
  + A white background with black text

    Description automatically generated
  + I have implemented separate action files for all the steps of CI and called those steps in the workflow file named hackathon-CI
* **Helm Charts Deployment**
* I have designed the Helm charts to deploy it seamlessly in multiple environments by just updating the values.yaml
* The files you can find inside the helm folder in the repo
* **CD Workflow**
* I have designed the CD workflow to Deploy the helm chart in AKS with my own strategy.
* Written one shell script to check for the latest pushed image value into the Azure container Registry and update the value in values.yaml
* So that it can deploy the charts using Helm upgrade –install command
* **Terraform to deploy Azure VM**
* I have written a terraform script to deploy an Azure Virtual machine
* Ansible SetUp to install MongoDB
* Designed Ansible playbook to install MongoDB

**.github tree view:**



**Helm Structure:**

