

Step-1:

First I created a student account to get the azure student services

The screenshot shows the Microsoft Azure Home page. At the top, there's a navigation bar with links like Chrome, File, Edit, View, History, Bookmarks, Profiles, Tab, Window, Help, and a search bar. Below the navigation bar, there are several tabs open in the browser, including YouTube, HTML Reference, CI/CD, AzmayenSabil/CICD-Lab..., Login | Microsoft 365, and Home - Microsoft Azure.

The main content area features a "Welcome to Azure!" message and a note about not having a subscription, followed by three promotional cards:

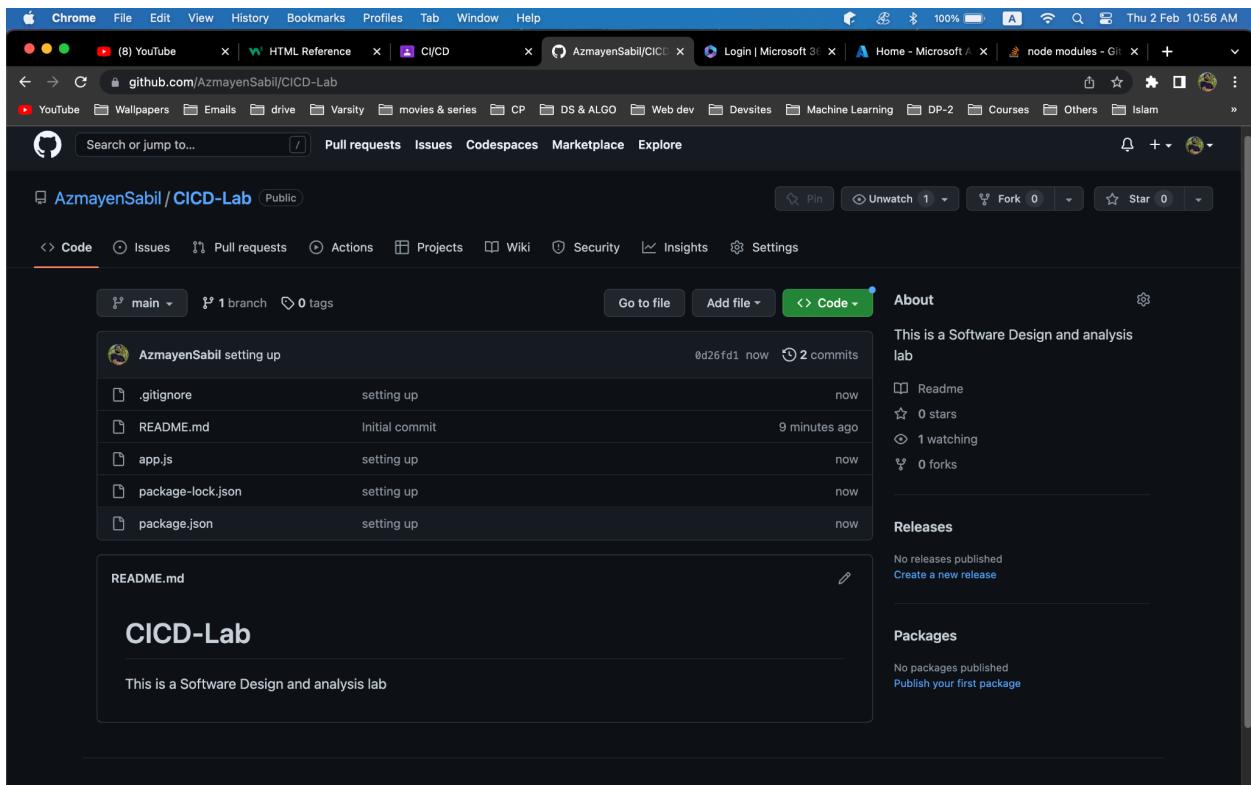
- Start with an Azure free trial**: Get \$200 free credit toward Azure products and services, plus 12 months of popular free services. Includes a "Start" button.
- Manage Azure Active Directory**: Manage access, set smart policies, and enhance security with Azure Active Directory. Includes a "View" button.
- Access student benefits**: Get free software, Azure credit, or access Azure Dev Tools for Teaching after you verify your academic status. Includes an "Explore" button.

Below these cards, there's a section titled "Azure services" with icons for Create a resource, Quickstart Center, Virtual machines, App Services, Storage accounts, SQL databases, Azure Cosmos DB, Kubernetes services, Function App, and More services.

At the bottom, there's a "Resources" section with a progress bar indicating "Waiting for js.monitor.azure.com...".

Step:2

After that I created a git repo based on nodeJS.



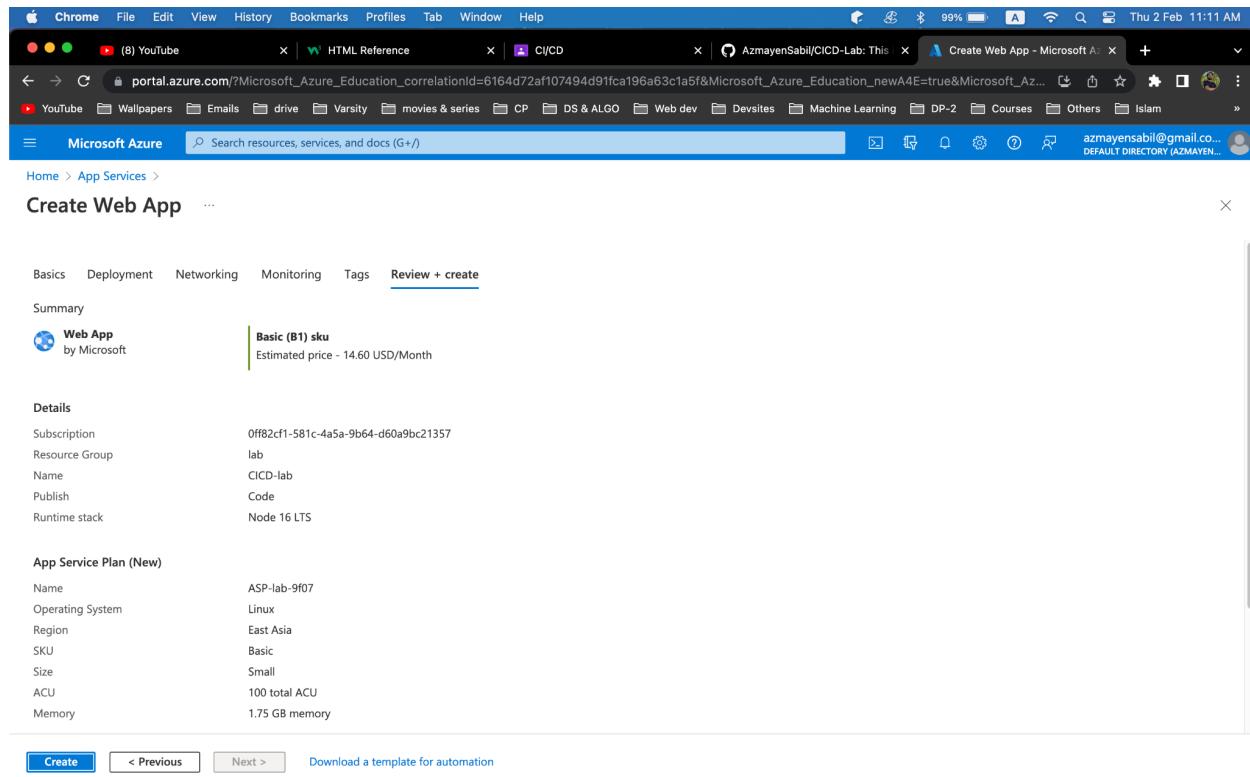
Step-3

After getting the approval from Azure with my student email I will be at this page.

The screenshot shows a Microsoft Azure Education Overview page in a Chrome browser. The URL in the address bar is https://portal.azure.com/?Microsoft_Azure_Education_correlationId=6164d72af107494d91fca196a63cta5f&Microsoft_Azure_Education_newA4E=true&Microsoft_Az.... The page title is "Education | Overview". The main content area includes sections for "Student offer details" (Available credits: \$100 out of \$100, Days until credits expire: 366), "Popular solutions" (Deploy a Docker container, Create your first Node.js app, Create and train a Machine Learning model, Build and deploy your first website), "Free Services" (Azure Virtual Machines – Windows, Azure Blob Storage, Computer Vision, Azure App Service), and "Resources" (Get started guide for Azure developers, Pricing calculator, Optimize your cloud investment with cost ...). On the left sidebar, there are sections for "Learning resources" (Roles, Software, Learning, Templates) and "Need help?" (Support).

Step-4

Then we will go to Home and create a web app from the app service with necessary information such as Node version, operating system, Server location etc.



The screenshot shows a Microsoft Azure portal page titled "Create Web App". The page is divided into several sections:

- Summary:** Shows a "Web App by Microsoft" icon and a "Basic (B1) sku" section indicating an estimated price of 14.60 USD/Month.
- Details:** Lists the following configuration:
 - Subscription: 0ff82cf1-581c-4a5a-9b64-d60a9bc21357
 - Resource Group: lab
 - Name: CICD-lab
 - Publish: Code
 - Runtime stack: Node 16 LTS
- App Service Plan (New):** Lists the following configuration:
 - Name: ASP-lab-9f07
 - Operating System: Linux
 - Region: East Asia
 - SKU: Basic
 - Size: Small
 - ACU: 100 total ACU
 - Memory: 1.75 GB memory

At the bottom, there are buttons for "Create", "< Previous", "Next >", and "Download a template for automation".

Step-5

Then we will review and create and the deployment progress will be running.

The screenshot shows a Microsoft Azure web application overview page. The title bar indicates the URL is `portal.azure.com?Microsoft_Azure_Education_correlationId=6164d72af107494d91fca196a63c1a5f&Microsoft_Azure_Education_newA4E=true&Microsoft_Az...`. The main content area is titled "Microsoft.Web-WebApp-Portal-49961a75-9d87 | Overview". A sidebar on the left shows navigation links for Overview, Inputs, Outputs, and Template. The main content area displays a message: "Deployment is in progress". Below this, deployment details are listed: Deployment name: Microsoft.Web-WebApp-Portal-49961a75-9d87, Start time: 2/2/2023, 11:12:33 AM, Subscription: Azure for Students, Correlation ID: e9bff33c-edb9-4faa-adb7-e83b305f056d. A table titled "Deployment details" shows one row: "No results.". At the bottom, there are links for "Give feedback" and "Tell us about your experience with deployment". To the right of the main content, there are promotional links for Microsoft Defender for Cloud, Microsoft tutorials, and Azure experts.

Step-6

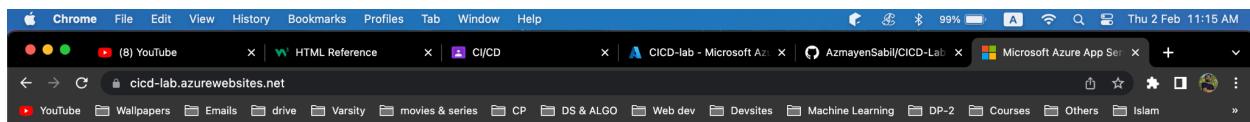
After the process is done we will get this interface.

The screenshot shows a Microsoft Azure portal page for a 'Web App' named 'CICD-lab'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment (Quickstart, Deployment slots, Deployment Center), Configuration, Authentication, Application Insights, Identity, Backups, and Custom domains. The main content area displays the 'Overview' tab for the 'CICD-lab' web app. It shows the following details:

Setting	Value
Resource group	(move) : lab
Status	: Running
Location	(move) : East Asia
Subscription	(move) : Azure for Students
Subscription ID	: 0ff82cf1-581c-4a5a-9b64-d60a9bc21357
Tags	: Click here to add tags
Name	CICD-lab
Publishing model	Code
Runtime Stack	Node - 16-lts
Name	Enable Application Insights
Deployment logs	View logs
Last deployment	No deployments found

Step-7

Then we will get an URL and using that url we will get this demo website.



Your web app is running and waiting for your content

Your web app is live, but we don't have your content yet. If you've already deployed, it could take up to 5 minutes for your content to show up, so come back soon.



 Built with NodeJS

Haven't deployed yet?
Use the deployment center to publish code or set up continuous deployment.

Starting a new web site?
Follow our Quickstart guide to get a web app ready quickly.

[Deployment center](#)

[Quickstart](#)

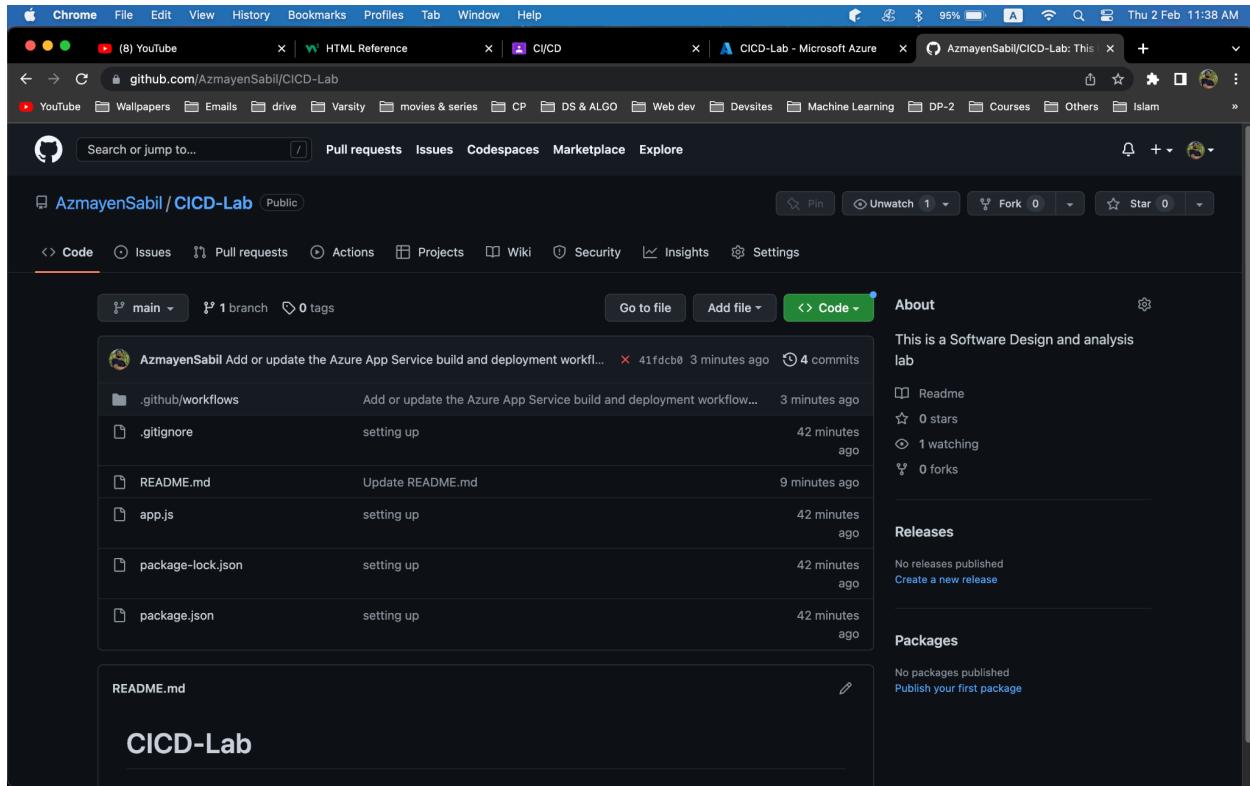
Step-8

Now we have to go to the deployment center and set up a few things like github account authorizing, repo selection and branch selection. Then just we have to save the environment.

The screenshot shows the Microsoft Azure portal interface. The browser tab is titled "CICD-Lab - Microsoft Azure". The address bar shows the URL "portal.azure.com/#/azmayensabil@gmail.onmicrosoft.com/resource/subscriptions/0ff82cf1-581c-4a5a-9b64-d60a9bc21357/resourcegroups/group/providers...". The main content area is titled "CICD-Lab | Deployment Center" and shows the "Settings" tab selected. The "Source" section is configured to use GitHub, with the repository set to "CICD-Lab" and the branch set to "main". The "Build" section shows "GitHub Actions" as the provider, "Node" as the runtime stack, and "Node 16 LTS" as the version. On the left sidebar, the "Deployment Center" option is highlighted under the "Deployment" category. The top navigation bar includes links for "Overview", "Logs", and "FTP credentials". The bottom right corner shows the user's email "azmayensabil@gmail.co..." and the status "DEFAULT DIRECTORY (AZMAYEN...)".

Step-9

After the previous step we will get a github/workflow folder inside our repo.



Step-10

We have to fix up a few things. First we have to use process.env.PORT to address our port.

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure for "CICD-LAB". It includes files like ".env", ".gitignore", "app.js", "package.json", and "README.md".
- Code Editor:** The "app.js" file is open, showing the following code:

```
const express = require('express')
const app = express()

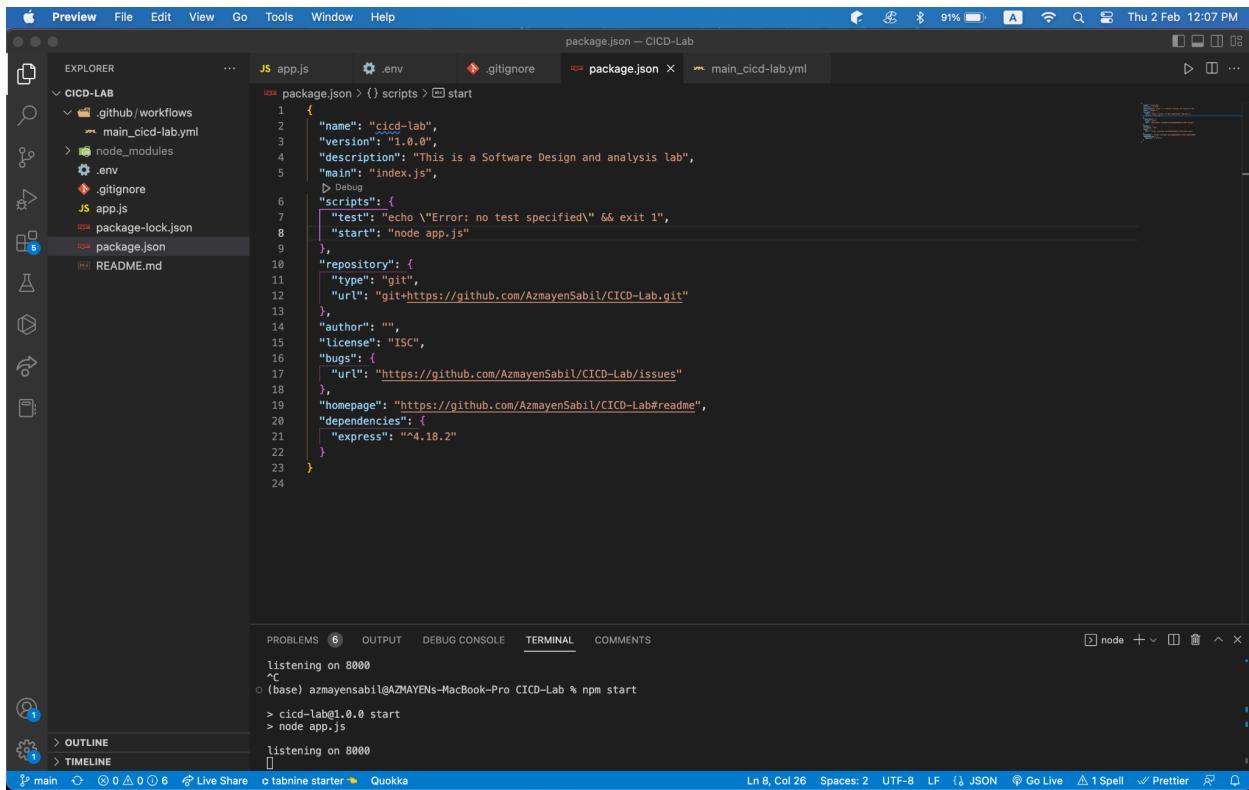
app.get('/', (req, res) => {
  res.send('Welcome Azure Application')
})

app.listen(process.env.PORT, () => {
  console.log('listening on 8000')
})
```
- Terminal:** The terminal window shows the command being run and its output:

```
listening on 8000
^C
(base) armayensil@AZMAYENS-MacBook-Pro CICD-Lab % npm start
> cicd-lab@1.0.0 start
> node app.js
listening on 8000
```
- Status Bar:** Shows the current file is "app.js - CICD-Lab", battery level at 93%, and the date/time as "Thu 2 Feb 11:56 AM".

Step-11

We have to edit the package.json with a new script "start" : "node app.js"



```
package.json — CICD-Lab
{
  "name": "cicd-lab",
  "version": "1.0.0",
  "description": "This is a Software Design and analysis lab",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "node app.js"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/AzmayenSabil/CICD-Lab.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/AzmayenSabil/CICD-Lab/issues"
  },
  "homepage": "https://github.com/AzmayenSabil/CICD-Lab#readme",
  "dependencies": {
    "express": "4.18.2"
  }
}
```

The screenshot shows the VS Code interface with the package.json file open in the center editor. The "start" script is highlighted with a pink selection. Below the editor, the terminal window shows the command "npm start" being run, followed by the output "listening on 8000". The status bar at the bottom indicates the file is 26 lines long and uses JSON syntax.

Step-12

We have to create a .env file to get the localhost address by using process.env.PORT.

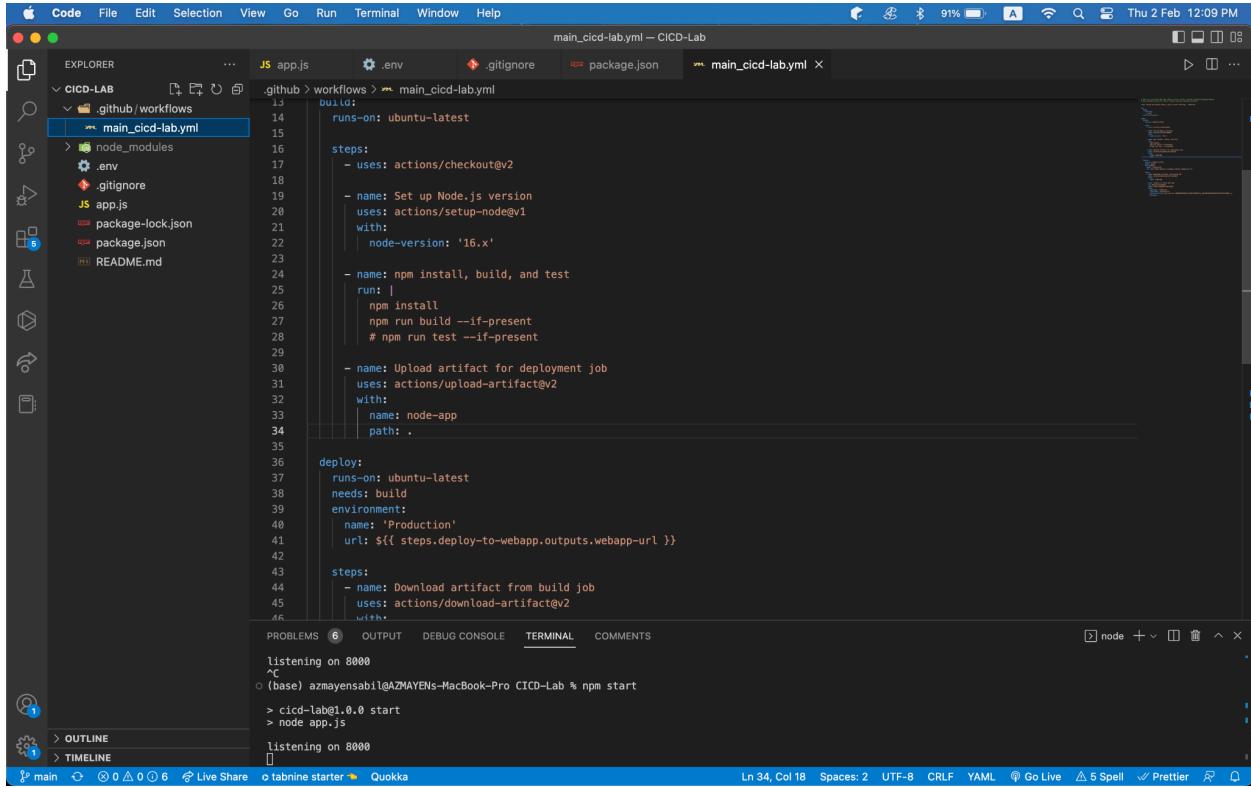
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project folder named "CICD-LAB" containing files: .github, node_modules, .env, .gitignore, app.js, package-lock.json, package.json, and README.md.
- Editor:** A .env file is open with the content: `1 PORT = 8000`.
- Terminal:** The terminal tab is active, showing the command `npm start` being run. The output shows the application starting on port 8000:

```
listening on 8000
^C
c:\base\azmayensabil@AZMAYENS-MacBook-Pro CICD-Lab % npm start
> cicd-lab1.0.0 start
> node app.js
listening on 8000
```
- Status Bar:** Shows the current file is "main", there are 6 problems, and the code editor is in Quokka mode.

Step-13

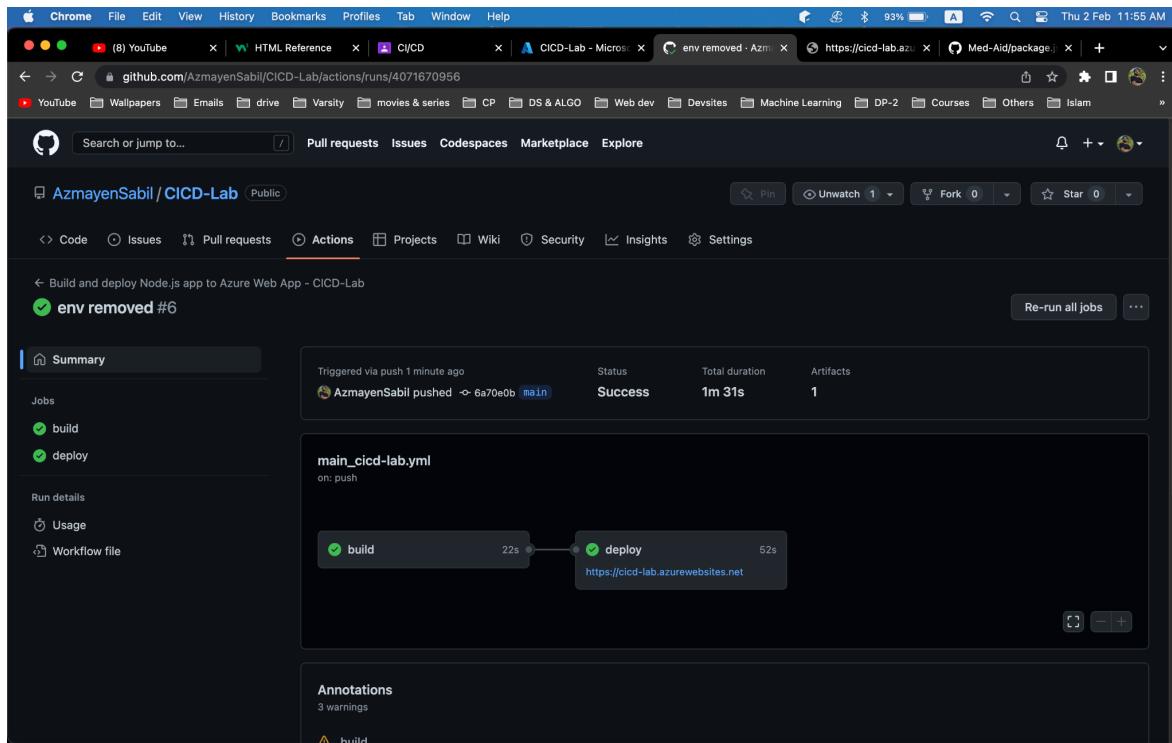
Another thing that we have to do is we have comment out npm run test from the yml file that was created after the azure deployment.



```
main_cicd-lab.yml — CICD-Lab
Code File Edit Selection View Go Run Terminal Window Help
EXPLORER GitHub workflows main_cicd-lab.yml
CICD-LAB .github workflows main_cicd-lab.yml
  node_modules
  .env
  .gitignore
  app.js
  package-lock.json
  package.json
  README.md
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
        name: Set up Node.js version
        uses: actions/setup-node@v1
        with:
          node-version: '16.x'
      - name: npm install, build, and test
        run: |
          npm install
          npm run build --if-present
          # npm run test --if-present
      - name: Upload artifact for deployment job
        uses: actions/upload-artifact@v2
        with:
          name: node-app
          path: .
  deploy:
    runs-on: ubuntu-latest
    needs: build
    environment:
      name: 'Production'
      url: ${{ steps.deploy-to-webapp.outputs.webapp-url }}
    steps:
      - name: Download artifact from build job
        uses: actions/download-artifact@v2
        with:
PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL COMMENTS
listening on 8000
^C
(base) azmayensabil@AZMAYENs-MacBook-Pro CICD-Lab % npm start
> cicd-lab@1.0.0 start
> node app.js
listening on 8000
Ln 34, Col 18 Spaces: 2 UTF-8 CRLF YAML Go Live △ 5 Spell Prettier
```

Step-14

Then we just have to commit and push to github.



Step-15

And finally we will be able to see our node project after the deployment is done. The completion of the deployment will give us a link that will take us to this page.

