DDeployment

Go to EC2 and click on **RUNNING INSTANCE**. Then click on **LAUNCH INSTANCE**. This means I am creating a cloud server where I can deploy my model and create an API out of it.

Search for Ubunto and select a Ubuntu free tier. In the next page click on instance type as fee tier eligible. Again click on **LAUNCH**.

Whenever a new instance is created for a new model its better that we create a new key pair. Choose create a new key pair from the drop down. Write ML DEMO TEST in the key pair name space. Click **DOWNLOAD THE KE PAIR**. From the downloads copy the key pair file and paste it in local folder where all ML model files are stored.

A key pair consists of public key that AWS stores and private key that you store. Together, they allow you to connect to our instance securely.

Now click on **LAUNCH INSTANCE.**

Edit and write a name of the instance.

Create a docker file in Spider. Name it as Dockerfile.

Write

FROM python 3.8 –slim-buster

This is import python 3.8 version of Linus base image from Docker hub.

WORKDIR /app

COPY . /app

Here the work directory name app is created and copy the entire project in spider in to app directory.

Run apt update -y && apt install awscli -y

The above command is for updating all packages before the installation of requirement.txt

CMD[“python3” , “app.py”]

This the command to run the app.py

The work is being done in visual basisc or spider.

Go down and click on terminal and then cmd.

Now, in terminal tab as cmd.

docker images

Now, main.yaml file has the entire CICD pipeline.

[[ Whenever any push is happening the readme file will be ignored and rest files will be pushed for deploment. In the integration, unit test are run, checks codes in GITHUB. Linting is the automated checking of your source code for programmatic and stylistic errors. This is done by using a lint tool (otherwise known as linter). A lint tool is a basic static code analyzer. Unit tests - To ensure isolated functions / methods behave as expected. Integration tests - To ensure that a chain of actions (i.e function/method calls) behave as expected. Linters - To ensure that code is written in a uniform way (i.e. structure, spacing etc.) ]]

Now we build a docker image and push it to docker hub. Now in docker hub the image is public. For working with company we need private image. ECR in AWS is a place to store private docker image. And from ECR we can deploy the docker image EC2 instance.

Until the integration is successful and complete we will not push the image in ECR.

Build and push ECR image

Continuous Delivery – checks codes in Github , install pakages, establish or configure AWS credentials and secreate access keys and all. Now, push the image. And do the continuous deployment with same steps as continuous delivery. Next pull the image from ECR and deploy in EC2 instance.

YALM aws alread existing in Github action. Go to Action and click on New Workflow and you see “DEPLOY TO AMAZON ECS” . Click on configure. You will get YALM file.

Commit the entire code in VB in Github.

Now, create IAM user.

Go to user.

Add User.

Write user name

Click on “Provide user access to AWS management console”

Click Next

Permission option – Click on “Attach policies directly”

Search Amazon EC2 Container and select on Full Access

Do same for AWS EC2 Full Access

Click on Next and it will take you to summary of permission policy

Go to Securit credential tab and go ahead and create the access keys.

Click on CLI and next. Create Access key and download .csv file

Go to ECR and create a new repository. Set the visibility setting as private and put th ECR URL name as Project name or any name you want. Now click on create registory. Now the repository is created and we copy the URL of the registry and note it on a note pad.

Now we go to EC2 instance and create a virtual server in the cloud.

Launch instance and write the name of web server as project name

Select Ubuntu tier medium

Provide a name to ke pair or select an old key pair name or create a new key pair

Click on allow all HTTP traffic

Configure storage with increased GB as much as you want

Launch instance

Now,

Click on instance id once it is created

Click on Connect

Again Click on connect to coonect to instance

Command promt opens up just like server in front of me

Write : sudo app-get update -y

And all packages will be installed and indexed properly

Sudo app-get upgrade

curl –fsSL\ https: // get.docker.com –o get-docker.sh

sudo sh get-docker.sh

sudo usermode -aG docker ubuntu

newgrp docker

docker

………

Wenever you commit canges in github, deployment takes place in EC2. Tere is something called Ruuner in Gitub acbtion .

Create new Runner.

Select Linux

**Docker -- Create and Deploy**

**Dependencies --** There are various classes and objects defined when writing code. Most of the time, these classes depend on other classes in order to fulfill their intended purpose. These classes knows the resources it needs and how to get them. Basically, the process of supplying a resource that a given piece of code requires. The required resource is called dependency.

In the package.json file, there is an object called dependencies and it consists of all the packages that are used in the project with its version number. So, whenever you install any library that is required in your project that library you can find it in the dependencies object.

**Configuration --**The most common configuration includes credentials to the database or an external service, the hostname of the deployed server, dynamic parameters, etc.

Good configuration management should allow the software to be executed in any environment without changing the code. It helps Ops to manage all the hassle settings.

**Docker Image --** Lets say we ave application. And so are dependencies and configurations. Now we want to build a docker image. Docker Image is nothing but a collection of images such as Linus Base Image [ imported from docker hub ] , Anaconda Image , Python Image , and other dependencies.

Now when we run a docker image , a container is created which i\\s nothing but an environment where the application is run as soon as Docker Image is Run.

**Installation --** Ensure Bios is enable in CPU in task manager control panel. Hyper V is enabled in Windows Feature and 64 Bit Window 11. Now go to get started and download and install docker. Also, sign up Docker hub thoug its optional.

Type Docker in CMD promt to see docker is installed. C:/users/Arijit/ docker

**How to build Docker Image --**

**Step 1 -- Build a simple Flask application [ for ello World ] in app.py file in VB :**

from flask import flask

import os

app = Flask(\_name\_)

@app.route(‘ / ’, methods = [‘GET’])

def (home) :

return “Hello World”

if \_name\_ = “ \_main\_” :

app.run(debug = True, host = “0.0.0.0” , port = 5000)

With 0.0.0.0 I will be able to access local address and local host address.

Now go to the terminal , of your choice , below and write

python app.py

You will see my application is running on <http://127.0.0.1:5000/>

In order to run this application I require 1 library that is Flask which is tere in my requirement.txt

Step 2 – Build the docker image

Ceate a docker file in VB as Dockerfile

In the Dockerfile we write :

FROM python: 3.8-alpine

COPY . /app

WORKDIR /app

RUN pip install –r requirements.txt

CMD python app.py

Ten go to terminal CMD and Write :

docker build –t welcome-app .

docker images

Run docker image as container :

Docker run –p 5000:5000 welcome app

So, we have 2 Ips. One IP is for the container IP in my machine and other IP is for the host machine. If we use container IP it will not run. So, we use either host IP or local host IP 0.0.0.0

***We ae accessing from the host port to the container port. First 5000 is the host port and next is the container port.***

docker ps [ [ We will see how many containers are running ]]

docker stop 1346947 [[ \To stop the container, use container ID no. ]]

We can see the container running in Docker Desktop and also the image in Docker Desktop.

**Step 3 – Deplo te Flask image in Docker Hub repository :**

docker login

If docker hub is signed in then cedentials are not reuired in the CMD.

To go by the naming convention in docker hub we will rename it. So, first we remove the image then again build the image with the user name and the docker image name.

docker image rm –f welcome-app

docker build –t [arijit17/](file:///\\arijit17/) welcome-app .

Again renaming it in another way

docker tag arijit17/welcome-app arijit17/welcome-app1

docker push arijit17/welcome-app : latest

Here tag is latest

**Step 4 – Pull the image in the local machine :**

docker pull arijit17/welcome-app : latest

**Concept of Docker Compose** **–** It is configuration file like yalm file where configuration for running multiple docker container and inter communication between container is scripted.

**IP – PORT – HOST – LOCAL HOST --** [[ Some IP addresses are reserved for certain reasons. When you send a request to a loopback address, this triggers a loopback, meaning the request is sent back to the server it came from. As a result, loopbacks don’t go through the internet — they stay in your local network.

"When it comes to computer networks, localhost is like a secret code that refers to the very computer you are using. Think of it as a way of saying "this computer." It serves a purpose when you make a special request to your own device, known as a loopback request. These requests are handy for testing and security purposes, as we'll explore later.Typically, you can access the localhost of any computer through a loopback address called 127.0.0.1. By default, this IP address points to a server that is running on the same device. So when your computer asks for the IP address 127.0.0.1, it's essentially making a request to itself, its "local host."

"If you type "localhost" into your browser, your request won't travel through the internet. Instead, it will loop back and end up right back on your computer.

When you're in the process of building a website, it's important to test how it looks and functions in a web browser. However, you might not want to make your unfinished website live and accessible to the public just yet. This is where localhost comes in handy.By hosting your website files on your personal computer and making them available via localhost, you can access your website through your web browser and simulate the experience of visiting it from a remote device. The best part is that everything stays on your computer and remains private.

"This is a common practice among application and web developers who use localhost as a private testing server for their websites and applications. It allows them to test programs without the need to send files through the internet, making it more secure and ensuring that their website is not exposed to the public before it's ready.

IP Address -- "To visit a website, you enter that website’s domain name into your browser bar — for example, hubspot.com.But, a domain name alone doesn't identify a website. Rather, a domain name is simply an alias for a website’s IP address, a combination of numbers that identifies a device on a network. An IP address is made up of four numbers, each between 0 and 255. For example, the IP address associated with the server that hosts HubSpot's website is 104.19.154.83. Every device has a unique IP address, even your PC.When you punch “hubspot.com” into your browser bar and hit enter, the Domain Name System (DNS) takes what you entered, sees that it's paired with the IP address 104.19.154.83, and routes your request to the right place, HubSpot’s web server. The reality is a bit more complicated, but that's the gist. ]]