Following the tutorial, Developing a RESTful API with Go and Gin

To deploy onto AWS, there are few options

1. Run the executable file directly on EC2. which you need to
    a. Cross-compile your code for AWS EC2 environment
    b. Upload your executable onto EC2 using SSH/userscript
    c. Run your executable on EC2 using SSH/userscript
2. Compile the file and run it on EC2.
    a. Set Up AWS EC2 environment to install GoLang
    b. Copy your code from your repository to EC2
    c. Compile and Run your code / Use go run command on EC2 via SSH
3. Run your program on EC2 via docker image.
    a. Prepare the build script for your docker image https://docs.docker.com/language/golang/build-images/
    b. Set up your EC2 for go language support and install Docker Community Edition and Docker Compose using SSH/userscript
    c. Copy your code and folder onto EC2 using SSH/userscript
    d. Run docker build command on EC2 using SSH/userscript
    e. Deploy using docker-compose up -d command
4. Run your program via docker image from AWS ECS / EKS
    a. Build your docker image
    b. Upload your docker image to remote repository
    c. Set up the AWS ECS/EKS cluster
    d. Run it
5. Run your program using AWS App Runner

Step by Step to run the Executable file directly on EC2

Step 1: Cross-compile the code for AWS EC2 environment

Note I haven't found any success yet trying the following code in Window Power Shell or Command Prompt

On your local machine, Open a bash terminal (in VS Studio Code for window, you can open a git-bash terminal).

Navigate to the directory of your code folder with your main file, Type the following

```
GOOS=linux GOARCH=arm64 go build -o <your desire name> main.go
```

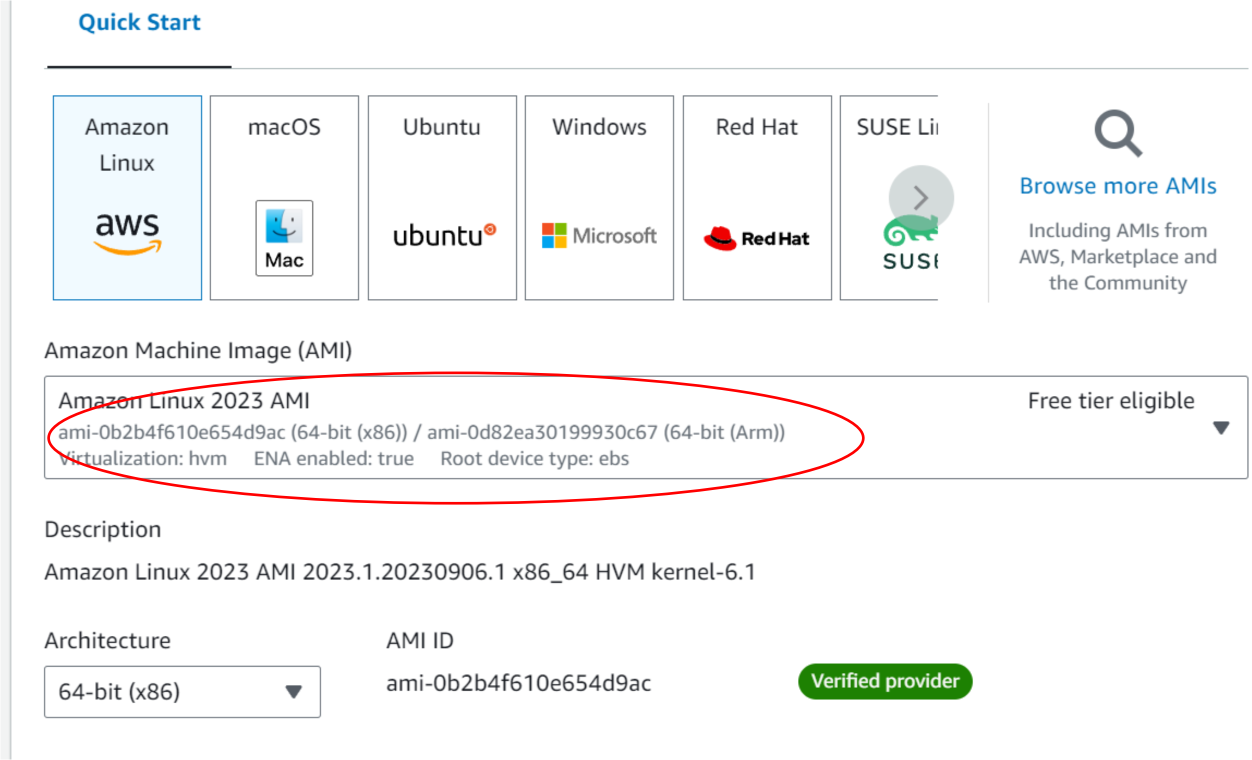
Here, GOOS set the target OS to linux, GOARCH set the target architecture to arm64, go build command build the executable file with code resides on main.go

Available target OS and compilation architecture as below

https://go.dev/doc/install/source#environment

For standard Amazon Linux 2023 AMI, will be looking at

GOOS – Linux , GOARCH – amd64 (amd64 is the most common one)



Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-0b2b4f610e654d9ac (64-bit (x86)) / ami-0d82ea30199930c67 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2023 AMI 2023.1.20230906.1 x86_64 HVM kernel-6.1

Architecture

64-bit (x86)

AMI ID

ami-0b2b4f610e654d9ac

Verified provider

Step 2: Copy your file to EC2

On your local machine command prompt

```
sudo scp -i <path to your pem file for aws private key, include.pem extension> <path to your executable binary cross-compile> ec2-user@<EC2_IP_ADDR>:<folder of your choices>
```

Note:

1. might need to use chmod R -777 <folder> command to give permission to upload file into the folder
2. For window, if you launch command prompt with administrator right, you can skip the sudo command

Step 3: Run your executable on EC2

Using SSH, navigate to the folder when you uploaded the executable binaries, simply type

```
./<your file name>
```

Note: might need to use chmod R -777 <file> command to give permission to run the file