MID SEMEMSTER

EVALUATION PROJECT

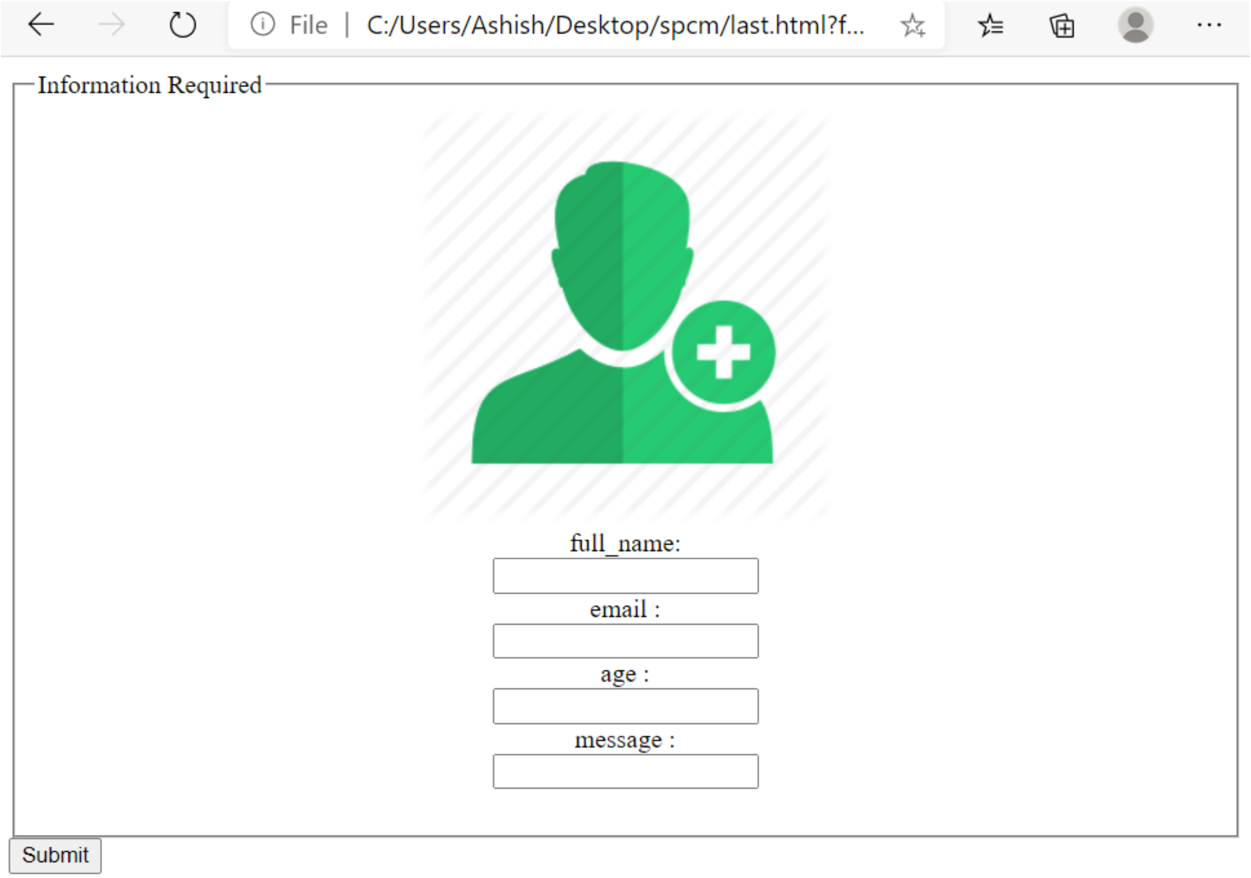
Name- Ashish Rajbhar

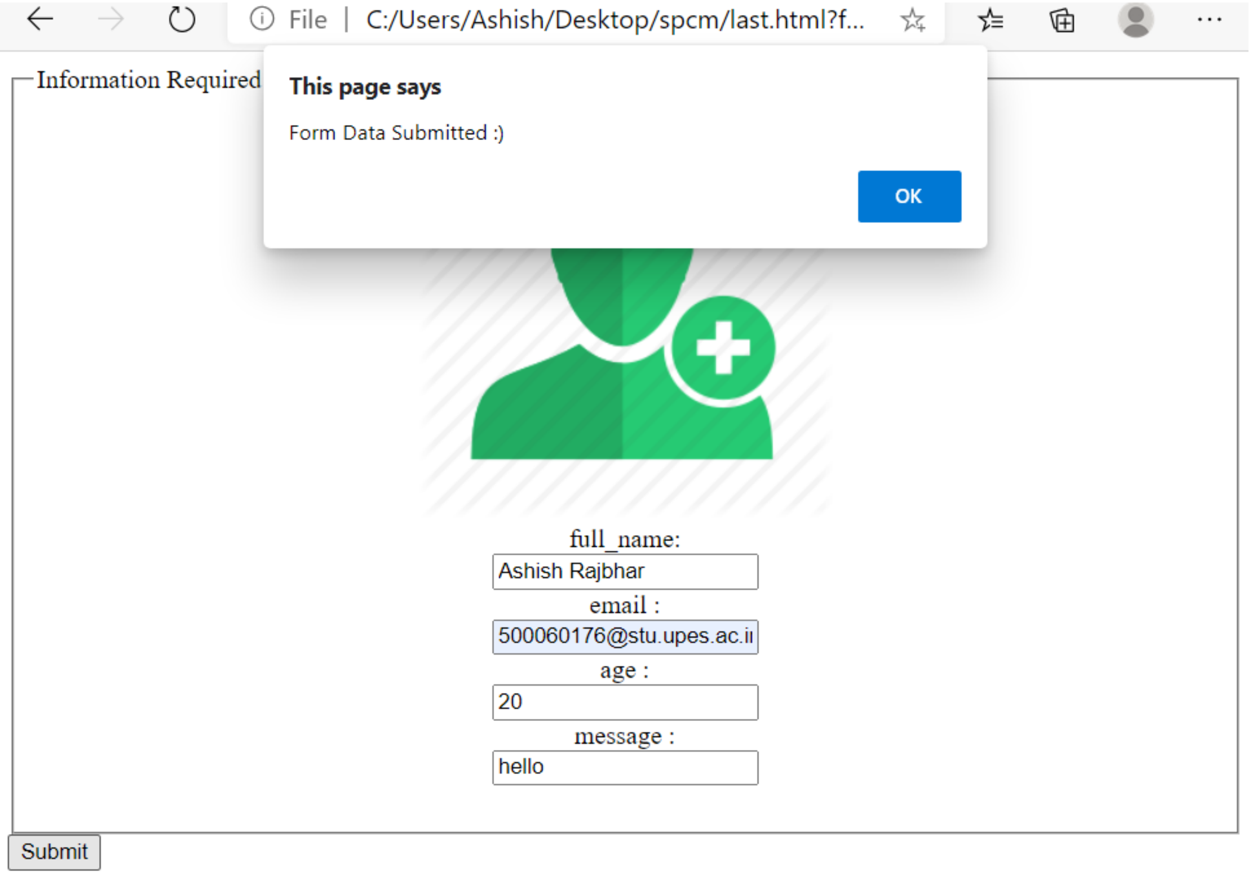
Sap ID- 500060176

Subject- SPCM

1. Create a small web application for storing information including Database (Google Spreadsheet).

   This application takes information and stores it in Google spreadsheet.





C:/Users/Ashish/Desktop/spcm/last.html?f...

This page says

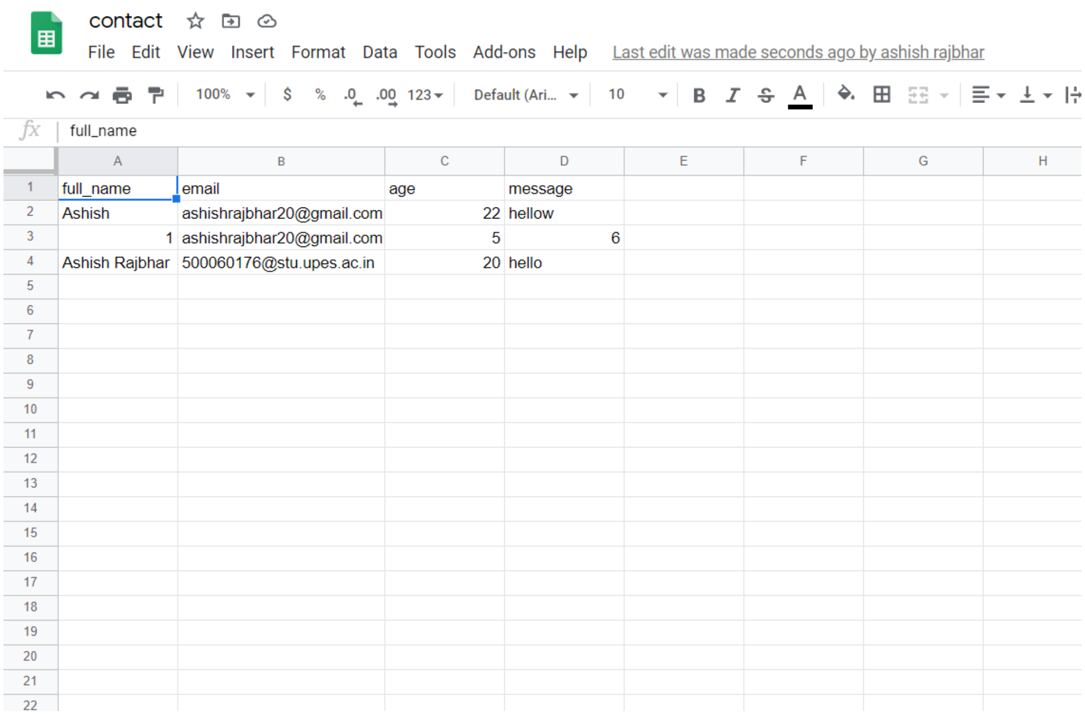
Form Data Submitted :)

full_name: Ashish Rajbhar

email : 500060176@stu.upes.ac.i

age : 20

message : hello



full_name

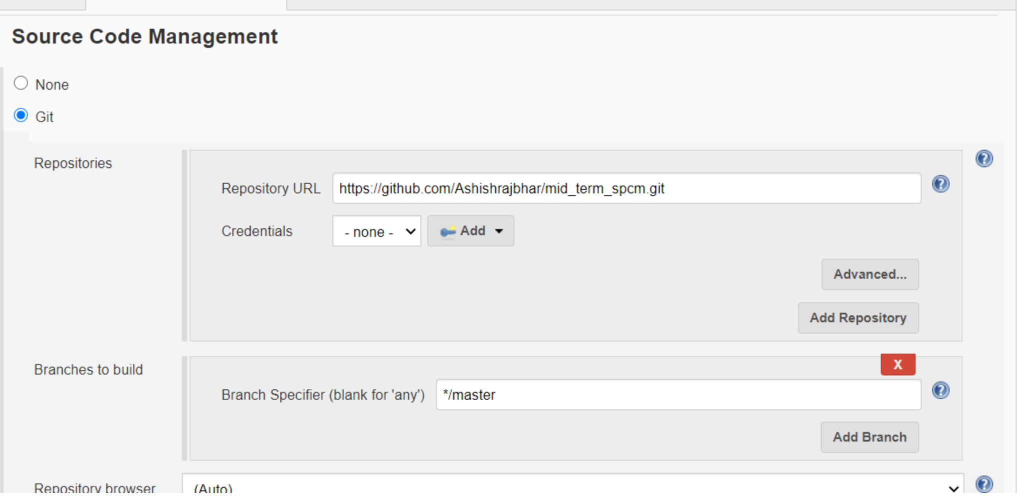
	A	B	C	D	E	F	G	H
1	full_name	email	age	message				
2	Ashish	ashishrajbhar20@gmail.com	22	hellow				
3	1	ashishrajbhar20@gmail.com	5	6				
4	Ashish Rajbhar	500060176@stu.upes.ac.in	20	hello				
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

2. Create a job in Jenkins to make build of this application.

Workflow: Jobs have been created.

Job 1: This job deals with fetching image from git repository, then building the image and at last deploying the built image to dockerhub repository.

Provide Jenkins the repository to fetch code from



Source Code Management

○ None

● Git

Repositories

Repository URL: https://github.com/Ashishrajbhar/mid_term_spcm.git

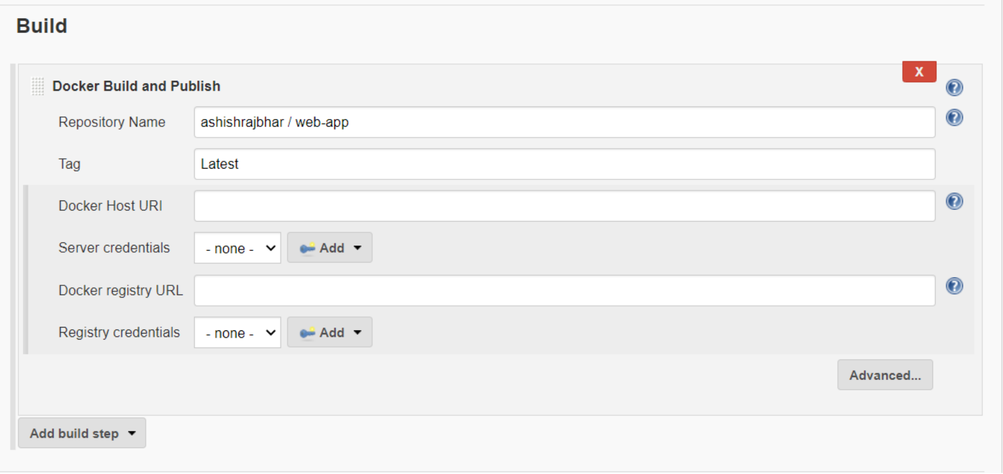
Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */master

Repository browser: (Auto)

Use the plugin to configure Jenkins to dockerhub repository



Build

Docker Build and Publish

Repository Name: ashishrajbhar / web-app

Tag: Latest

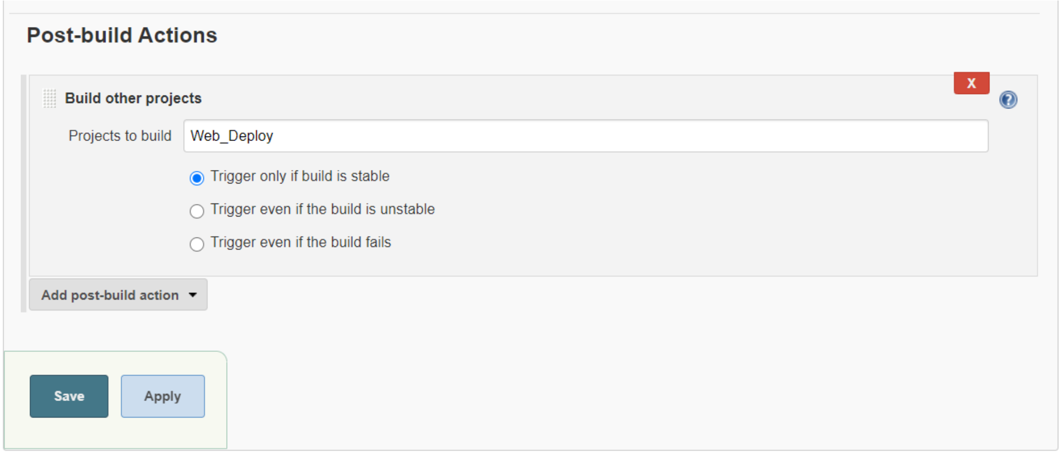
Docker Host URI:

Server credentials: - none -

Docker registry URL:

Registry credentials: - none -

Set the post build actions to trigger the deployment build



Post-build Actions

Build other projects

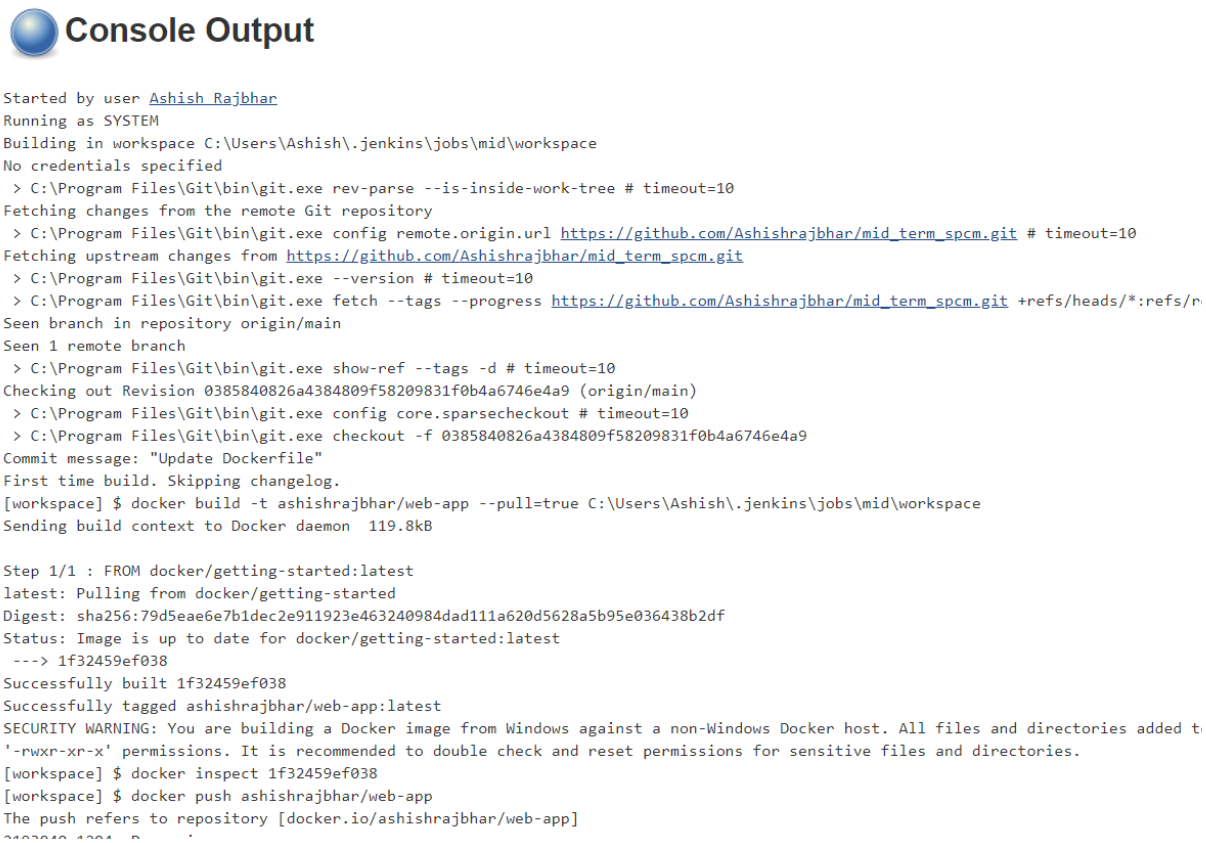
Projects to build: Web_Deploy

● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

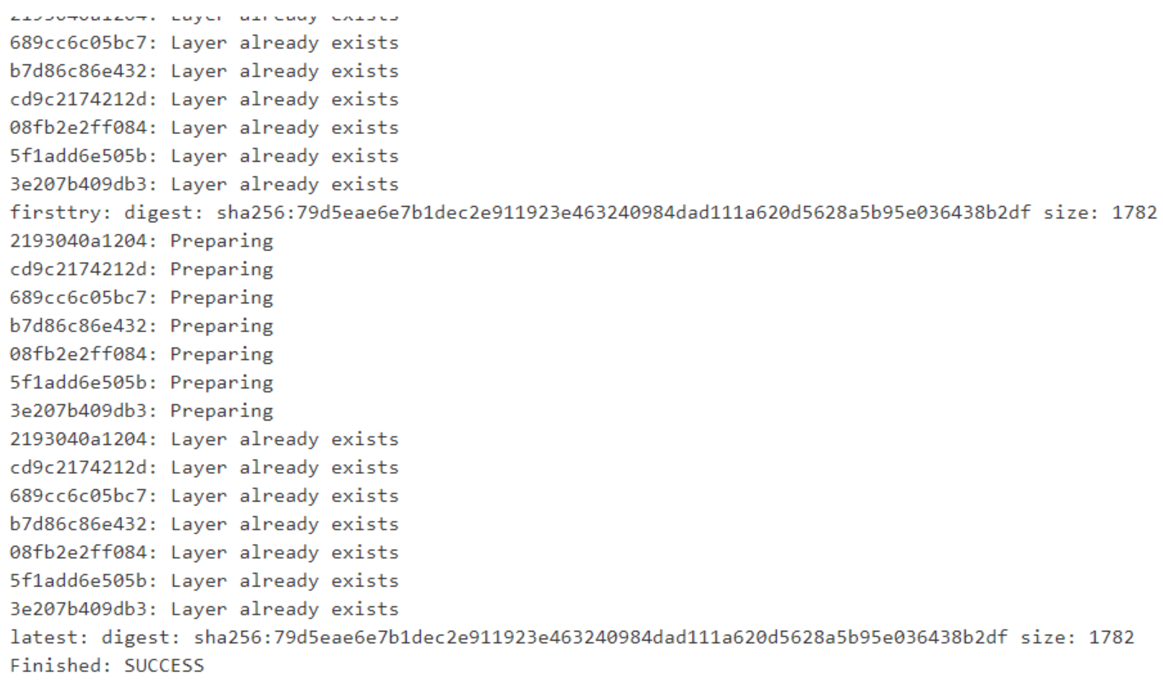
Console Output-



Console Output

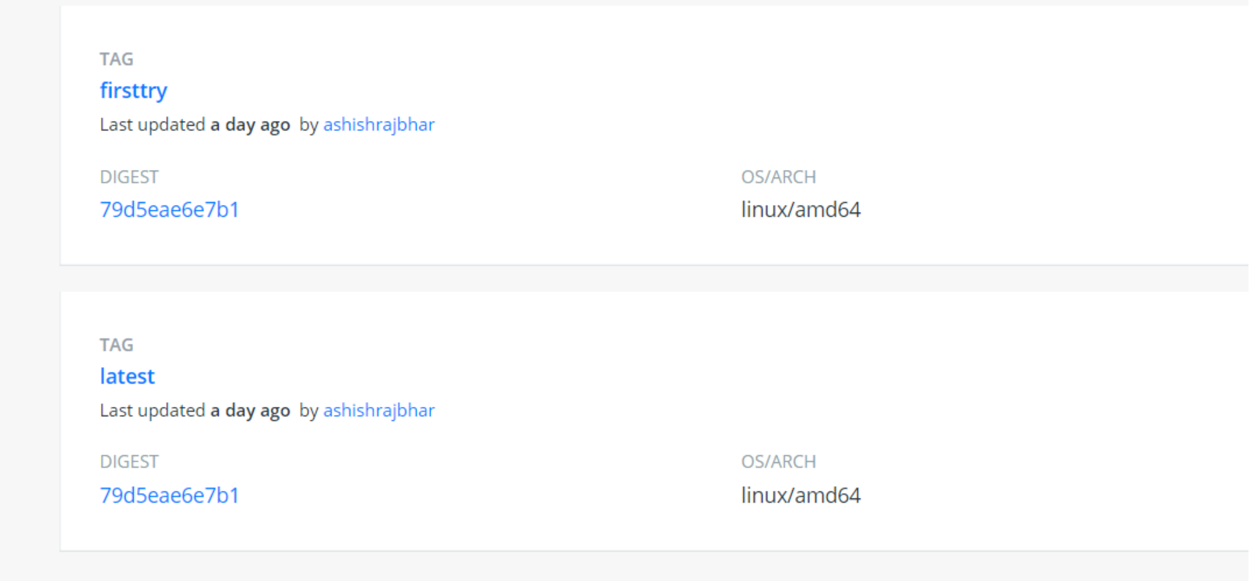
```
Started by user Ashish Rajbhar
Running as SYSTEM
Building in workspace C:\Users\Ashish\.jenkins\jobs\mid\workspace
No credentials specified
 > C:\Program Files\Git\bin\git.exe rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/Ashishrajbhar/mid_term_spcm.git # timeout=10
Fetching upstream changes from https://github.com/Ashishrajbhar/mid_term_spcm.git
 > C:\Program Files\Git\bin\git.exe --version # timeout=10
 > C:\Program Files\Git\bin\git.exe fetch --tags --progress https://github.com/Ashishrajbhar/mid_term_spcm.git +refs/heads/*:refs/r
Seen branch in repository origin/main
Seen 1 remote branch
 > C:\Program Files\Git\bin\git.exe show-ref --tags -d # timeout=10
Checking out Revision 0385840826a4384809f58209831f0b4a6746e4a9 (origin/main)
 > C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
 > C:\Program Files\Git\bin\git.exe checkout -f 0385840826a4384809f58209831f0b4a6746e4a9
Commit message: "Update Dockerfile"
First time build. Skipping changelog.
[workspace] $ docker build -t ashishrajbhar/web-app --pull=true C:\Users\Ashish\.jenkins\jobs\mid\workspace
Sending build context to Docker daemon  119.8kB

Step 1/1 : FROM docker/getting-started:latest
latest: Pulling from docker/getting-started
Digest: sha256:79d5eae6e7b1dec2e911923e463240984dad111a620d5628a5b95e036438b2df
Status: Image is up to date for docker/getting-started:latest
 ---> 1f32459ef038
Successfully built 1f32459ef038
Successfully tagged ashishrajbhar/web-app:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and directories added t
'-rwxr-xr-x' permissions. It is recommended to double check and reset permissions for sensitive files and directories.
[workspace] $ docker inspect 1f32459ef038
[workspace] $ docker push ashishrajbhar/web-app
The push refers to repository [docker.io/ashishrajbhar/web-app]
2193040a1204: Preparing
```



```
2193040a1204: Layer already exists
689cc6c05bc7: Layer already exists
b7d86c86e432: Layer already exists
cd9c2174212d: Layer already exists
08fb2e2ff084: Layer already exists
5f1add6e505b: Layer already exists
3e207b409db3: Layer already exists
firsttry: digest: sha256:79d5eae6e7b1dec2e911923e463240984dad111a620d5628a5b95e036438b2df size: 1782
2193040a1204: Preparing
cd9c2174212d: Preparing
689cc6c05bc7: Preparing
b7d86c86e432: Preparing
08fb2e2ff084: Preparing
5f1add6e505b: Preparing
3e207b409db3: Preparing
2193040a1204: Layer already exists
cd9c2174212d: Layer already exists
689cc6c05bc7: Layer already exists
b7d86c86e432: Layer already exists
08fb2e2ff084: Layer already exists
5f1add6e505b: Layer already exists
3e207b409db3: Layer already exists
latest: digest: sha256:79d5eae6e7b1dec2e911923e463240984dad111a620d5628a5b95e036438b2df size: 1782
Finished: SUCCESS
```

Build on DockerHub



TAG
firsttry
Last updated **a day ago** by ashishrajbhar

DIGEST
79d5eae6e7b1

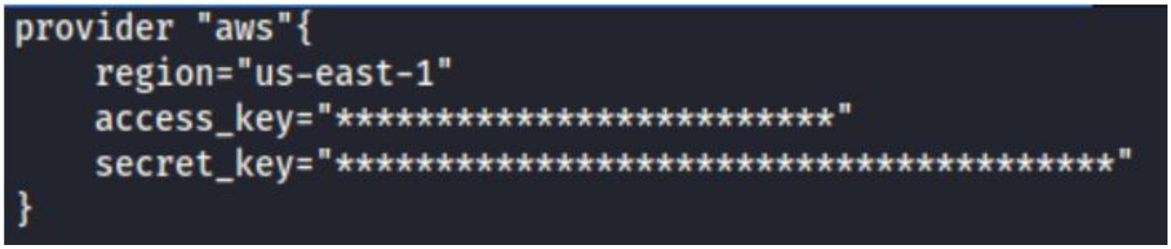
OS/ARCH
linux/amd64

TAG
latest
Last updated **a day ago** by ashishrajbhar

DIGEST
79d5eae6e7b1

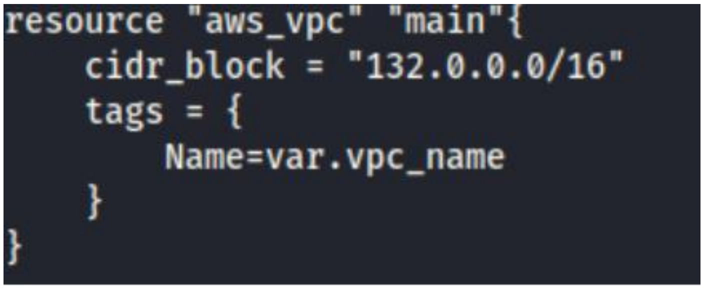
OS/ARCH
linux/amd64

1. Initialize the provider to set the API’s to be used. After setting “aws” as a provider you would be able to use the AWS resources.



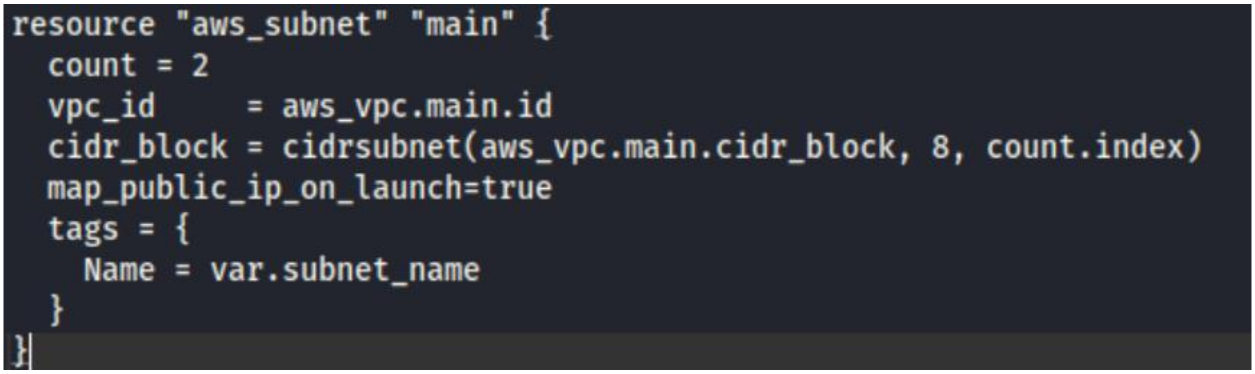
```
provider "aws"{
    region="us-east-1"
    access_key="*************************"
    secret_key="***************************************"
}
```

2. Next step is to set up the VPC. This is the place where we would be deploying our ECS cluster



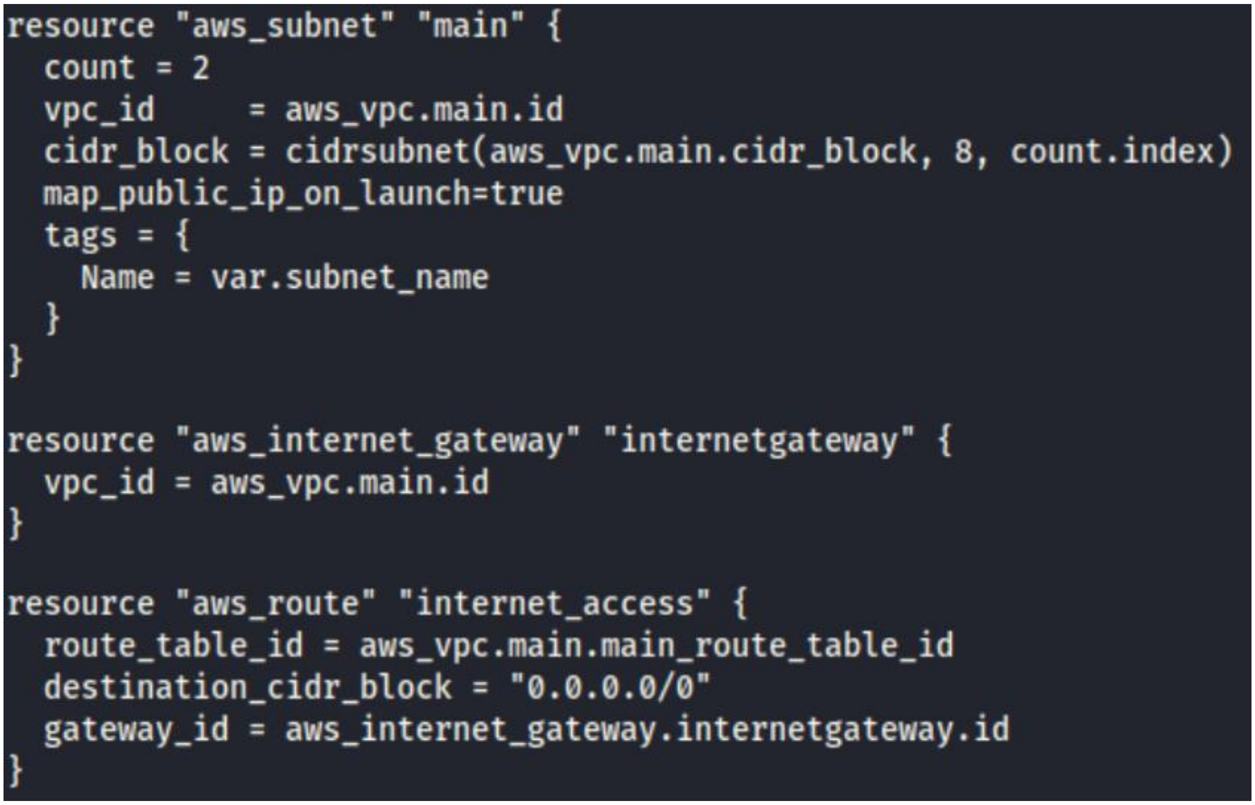
```
resource "aws_vpc" "main"{
    cidr_block = "132.0.0.0/16"
    tags = {
        Name=var.vpc_name
    }
}
```

3. After creating VPC we would have to setup the subnet configuration.



```
resource "aws_subnet" "main" {
  count = 2
  vpc_id     = aws_vpc.main.id
  cidr_block = cidrsubnet(aws_vpc.main.cidr_block, 8, count.index)
  map_public_ip_on_launch=true
  tags = {
    Name = var.subnet_name
  }
}
```

4. Next step involves configuration of our vpc and subnets.



```
resource "aws_subnet" "main" {
  count = 2
  vpc_id     = aws_vpc.main.id
  cidr_block = cidrsubnet(aws_vpc.main.cidr_block, 8, count.index)
  map_public_ip_on_launch=true
  tags = {
    Name = var.subnet_name
  }
}

resource "aws_internet_gateway" "internetgateway" {
  vpc_id = aws_vpc.main.id
}

resource "aws_route" "internet_access" {
  route_table_id = aws_vpc.main.main_route_table_id
  destination_cidr_block = "0.0.0.0/0"
  gateway_id = aws_internet_gateway.internetgateway.id
}
```

AFTER CREATING TERRAFORM CONFIG FILES RUN FOLLOWING COMMANDS ON THE TERMINAL:

1. terraform init
2. terraform plan
3. terraform apply