Assessment Theory

# Section 1: Cloud Platforms (AWS, Azure, GCP) (60 minutes)

# Explain how you would set up a highly available and scalable web application using AWS services. Include VPC, EC2, Load Balancer, Auto Scaling, and RDS in your explanation.

When we Talk About Highly Available and scalable web application using AWS services then we can use these Services.

1.VPC -> VPC Extends for Virtual Private Cloud. In this we need to take care of our Security so we can make subnet in private so any other person or attacker cannot attack on our application.

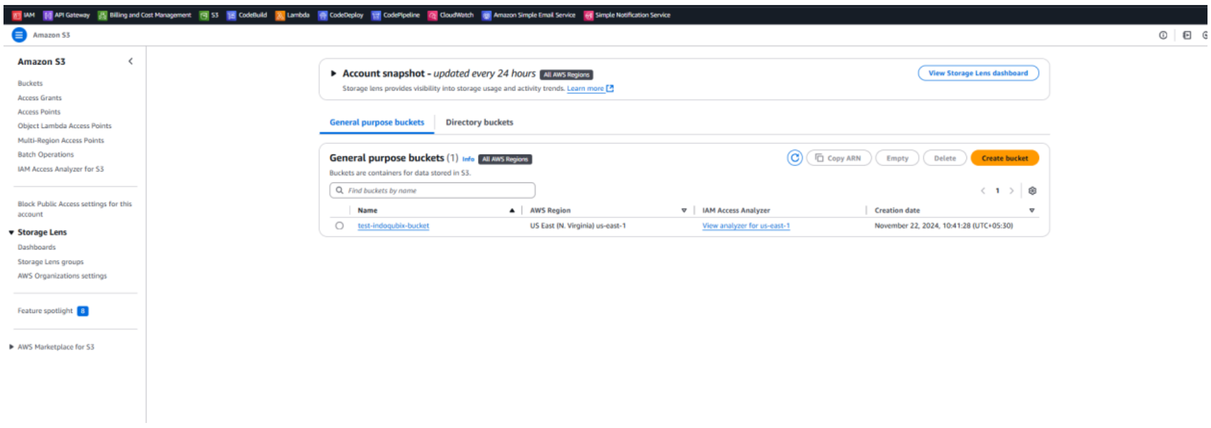
2.EC2 --> EC2 tends to Elastic Compute in which we can make our server in private subnet so we can access that server form own Network.

3.Loadbalancer --> Load Balancer Means Any Application can own due to high load or we can say due to no. of visit in that site so we can set load balancer on that so when the n. o fuser increases the its automatically redirection other server and replicate server.

4.Auto Scaling --> Auto Scaling is use to Scale your application or replicate your server whenever your server down.

5.RDS --> RDS tends to Relation Database in AWS to Make highly available and scalable web application we need to enable Multizone so we can application can access form anywhere easily.

https://test-indoqubix-bucket.s3.us-east-1.amazonaws.com/index.html



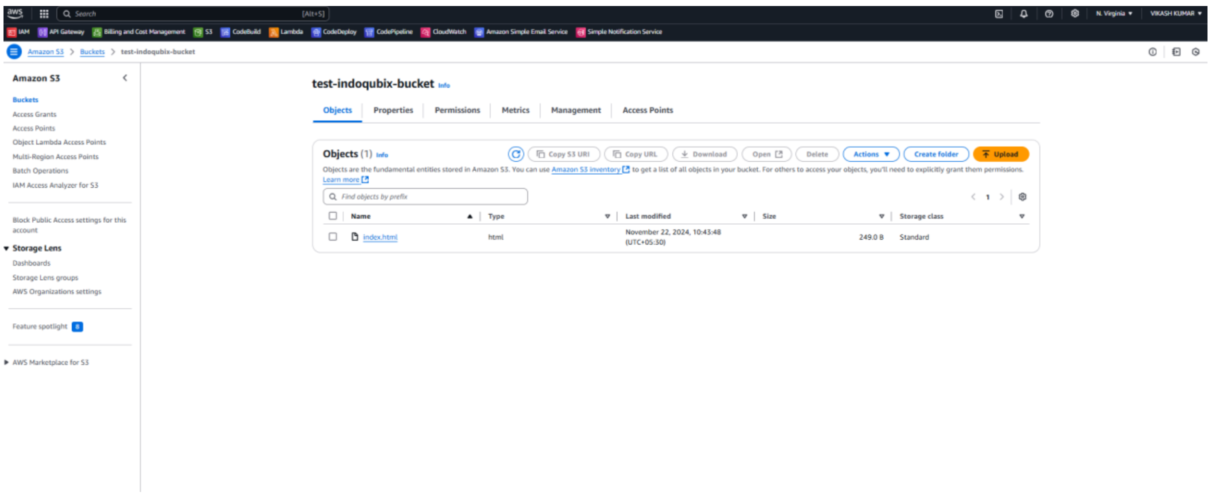
Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1)

Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
test-indoqubix-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	November 22, 2024, 10:41:28 (UTC+05:30)

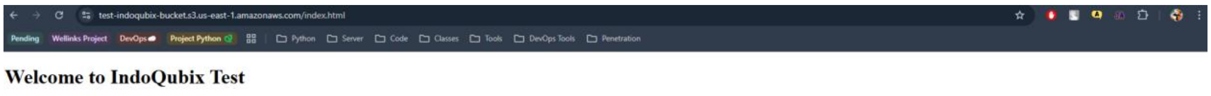


test-indoqubix-bucket

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
index.html	html	November 22, 2024, 10:43:48 (UTC+05:30)	249.0 B	Standard



test-indoqubix-bucket.s3.us-east-1.amazonaws.com/index.html

Welcome to IndoQubix Test

# Section 2: Infrastructure as Code (45 minutes)

54.185.106.202 → Public IP

Terraform:

# Write a Terraform script to deploy an EC2 instance in AWS with the following specifications: t2.micro, in the us-west-2 region, and with a specific security group allowing SSH access.

Created Terraform Script to Deploy an EC2 Instance.

Specifications

t2.micro

us-west-2

security group so we need to enable 22 Port

Code I Have Attached.

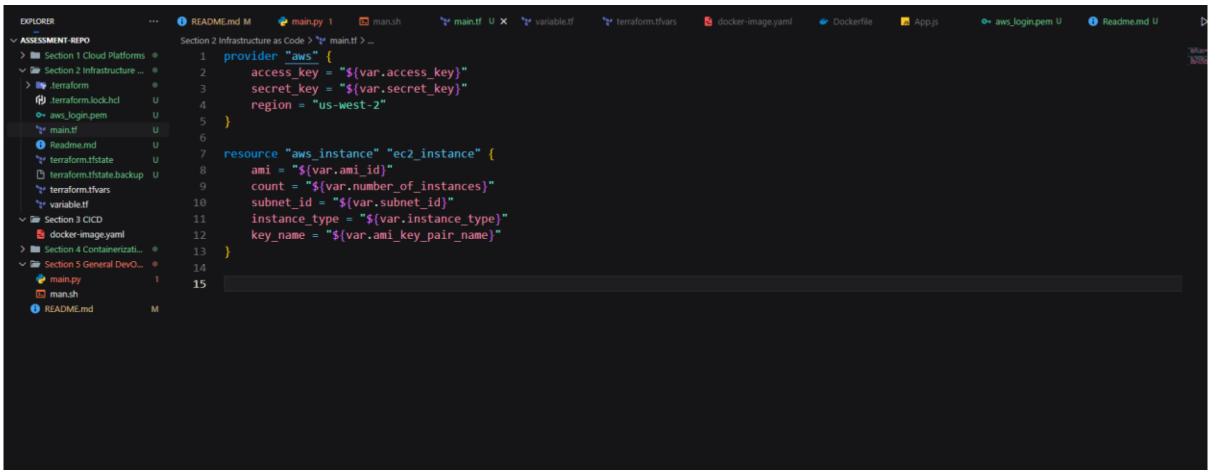
Terraform introduced by HCL

Using This Script, we can Make Any ENV in Any Cloud Provider.

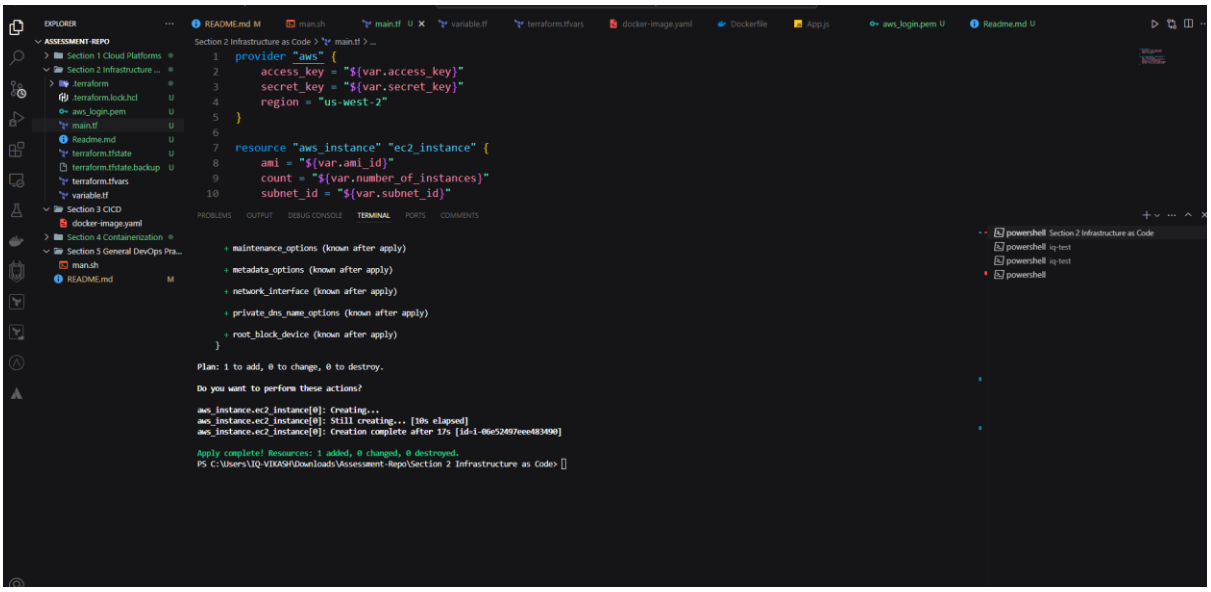
t2.micro --> Free Trail

us-west-2 --> We Need to Select Zone

security group so we need to enable 22 Port --> Enable Port to Do ssh



```
provider "aws" {
    access_key = "${var.access_key}"
    secret_key = "${var.secret_key}"
    region = "us-west-2"
}

resource "aws_instance" "ec2_instance" {
    ami = "${var.ami_id}"
    count = "${var.number_of_instances}"
    subnet_id = "${var.subnet_id}"
    instance_type = "${var.instance_type}"
    key_name = "${var.ami_key_pair_name}"
}
```



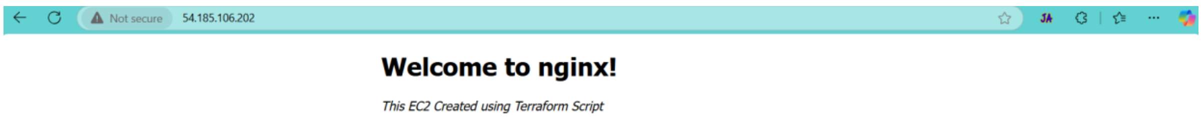
```
provider "aws" {
    access_key = "${var.access_key}"
    secret_key = "${var.secret_key}"
    region = "us-west-2"
}

resource "aws_instance" "ec2_instance" {
    ami = "${var.ami_id}"
    count = "${var.number_of_instances}"
    subnet_id = "${var.subnet_id}"
```

```
      + maintenance_options (known after apply)

      + metadata_options (known after apply)

      + network_interface (known after apply)

      + private_dns_name_options (known after apply)

      + root_block_device (known after apply)
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?

aws_instance.ec2_instance[0]: Creating...
aws_instance.ec2_instance[0]: Still creating... [10s elapsed]
aws_instance.ec2_instance[0]: Creation complete after 17s [id=i-06e52497eee483490]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\IQ-VIKASH\Downloads\Assessment-Repo\Section 2 Infrastructure as Code> 
```

To Run These, Terraform Code

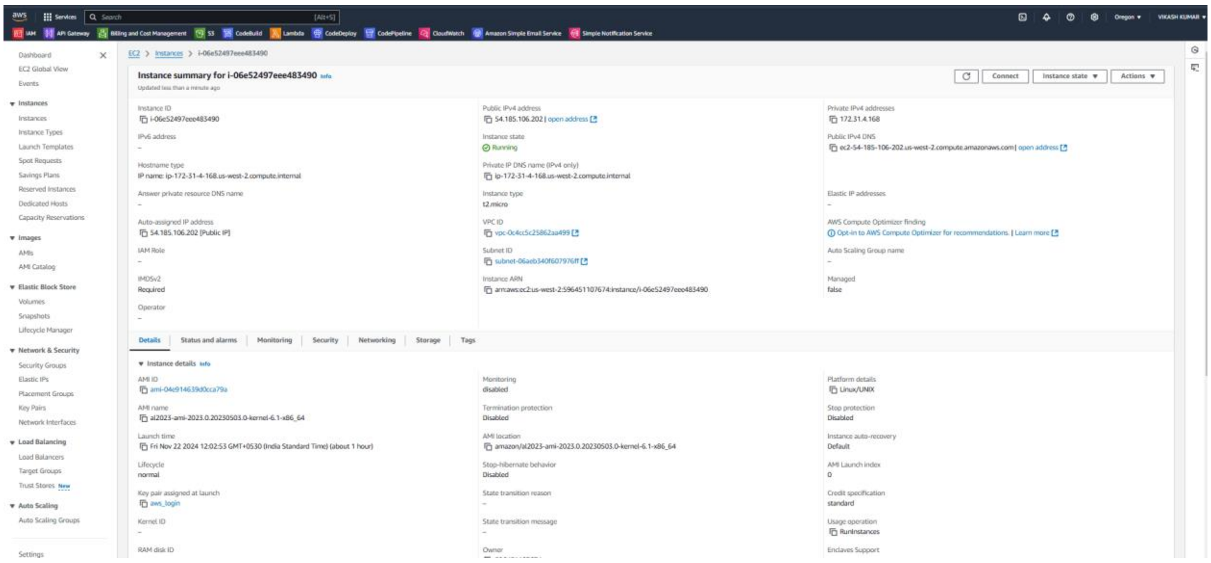
You need to install Terraform

- Terraform init
- Terraform plan
- Terraform apply



Welcome to nginx!

This EC2 Created using Terraform Script



Instance summary for i-06e52497eee483490 Info
Updated less than a minute ago

Instance ID
i-06e52497eee483490
IPv6 address
–
Hostname type
IP name: ip-172-31-4-168.us-west-2.compute.internal
Answer private resource DNS name
–
Auto-assigned IP address
54.185.106.202 [Public IP]
IAM Role
–
IMDSv2
Required
Operator
–

Public IPv4 address
54.185.106.202 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-172-31-4-168.us-west-2.compute.internal
Instance type
t2.micro
VPC ID
vpc-0c4cc5c25862aa499
Subnet ID
subnet-06aeb340f607976ff
Instance ARN
arn:aws:ec2:us-west-2:596451107674:instance/i-06e52497eee483490

Private IPv4 addresses
172.31.4.168
Public IPv4 DNS
ec2-54-185-106-202.us-west-2.compute.amazonaws.com | open address
Elastic IP addresses
–
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Auto Scaling Group name
–
Managed
false

Instance details Info

AMI ID
ami-04e914639d0cca79a
AMI name
al2023-ami-2023.0.20230503.0-kernel-6.1-x86_64
Launch time
Fri Nov 22 2024 12:02:53 GMT+0530 (India Standard Time) (about 1 hour)
Lifecycle
normal
Key pair assigned at launch
aws_login
Kernel ID
–
RAM disk ID

Monitoring
disabled
Termination protection
Disabled
AMI location
amazon/al2023-ami-2023.0.20230503.0-kernel-6.1-x86_64
Stop-hibernate behavior
Disabled
State transition reason
–
State transition message
–
Owner

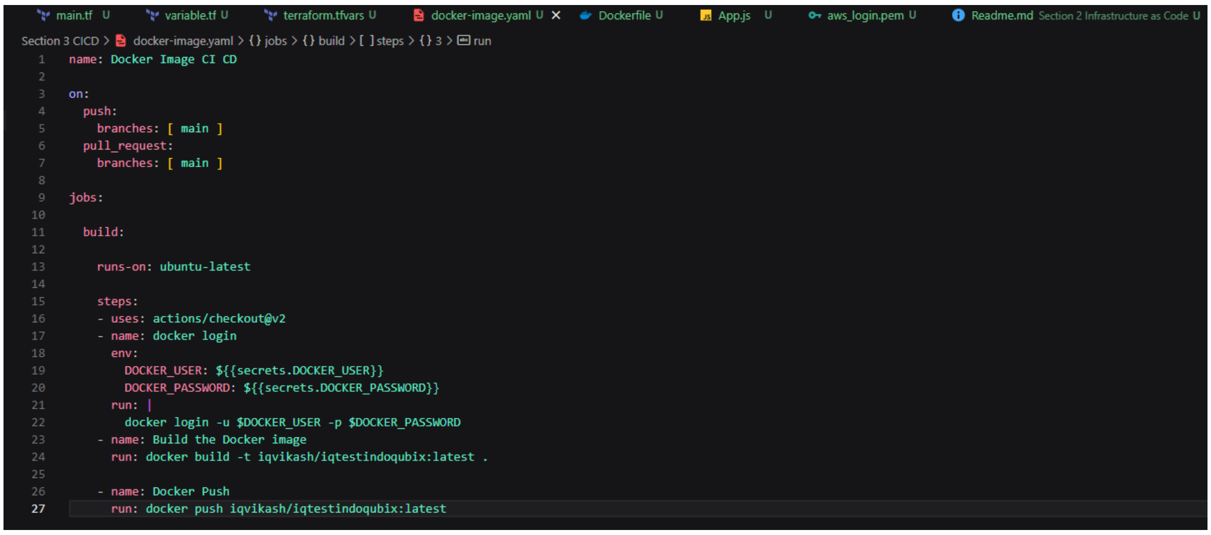
Platform details
Linux/UNIX
Stop protection
Disabled
Instance auto-recovery
Default
AMI Launch index
0
Credit specification
standard
Usage operation
RunInstances
Enclaves Support

# Section 3: CI/CD (Continuous Integration and Continuous Deployment) (45 minutes)

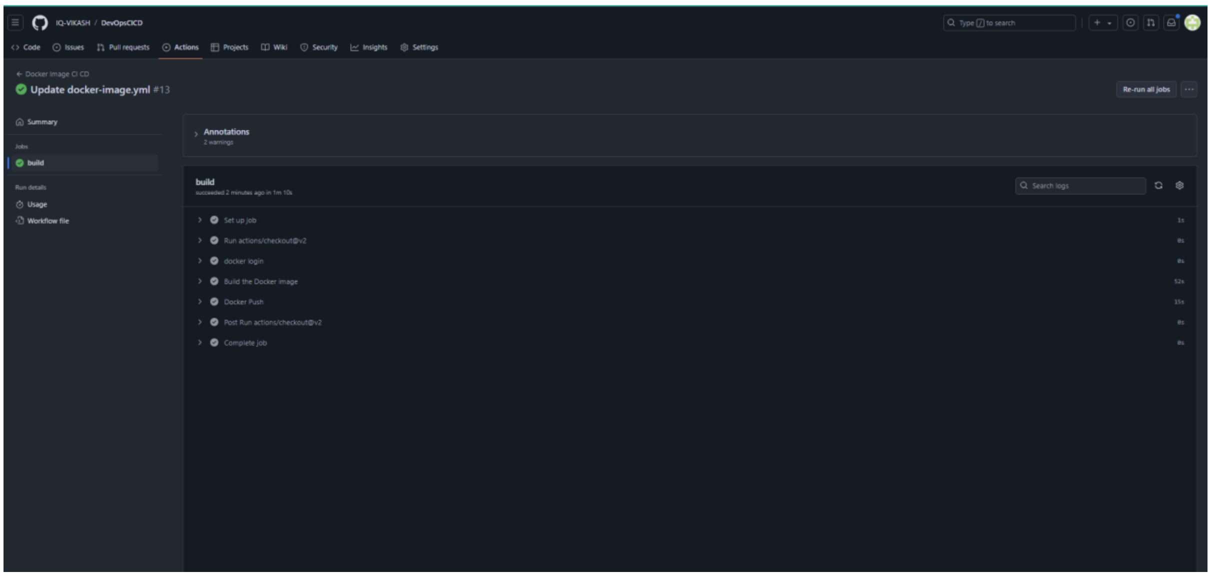
# Write a GitHub Actions workflow file to build a Docker image from a Docker file, run tests inside the container, and push the image to Docker Hub.

I Have attached the yaml workflow file in which I have write how the flow start

- 1.Create Repo for Project (https://github.com/IQ-VIKASH/DevOpsCICD/blob/main/.github/workflows/docker-image.yml)
- 2.Create Docker file for the project I have push to GitHub.
- 3.Also set the password in Secrets of Repo.
- 4.Start your Workflow.


main.tf U
variable.tf U
terraform.tfvars U
docker-image.yaml U
Dockerfile U
App.js U
aws_login.pem U
Readme.md Section 2 Infrastructure as Code U
Section 3 CICD > docker-image.yaml > {} jobs > {} build > [ ] steps > {} 3 > run

```yaml
name: Docker Image CI CD

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

jobs:

  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v2
    - name: docker login
      env:
        DOCKER_USER: ${{secrets.DOCKER_USER}}
        DOCKER_PASSWORD: ${{secrets.DOCKER_PASSWORD}}
      run: |
        docker login -u $DOCKER_USER -p $DOCKER_PASSWORD
    - name: Build the Docker image
      run: docker build -t iqvikash/iqtestindoqubix:latest .

    - name: Docker Push
      run: docker push iqvikash/iqtestindoqubix:latest
```



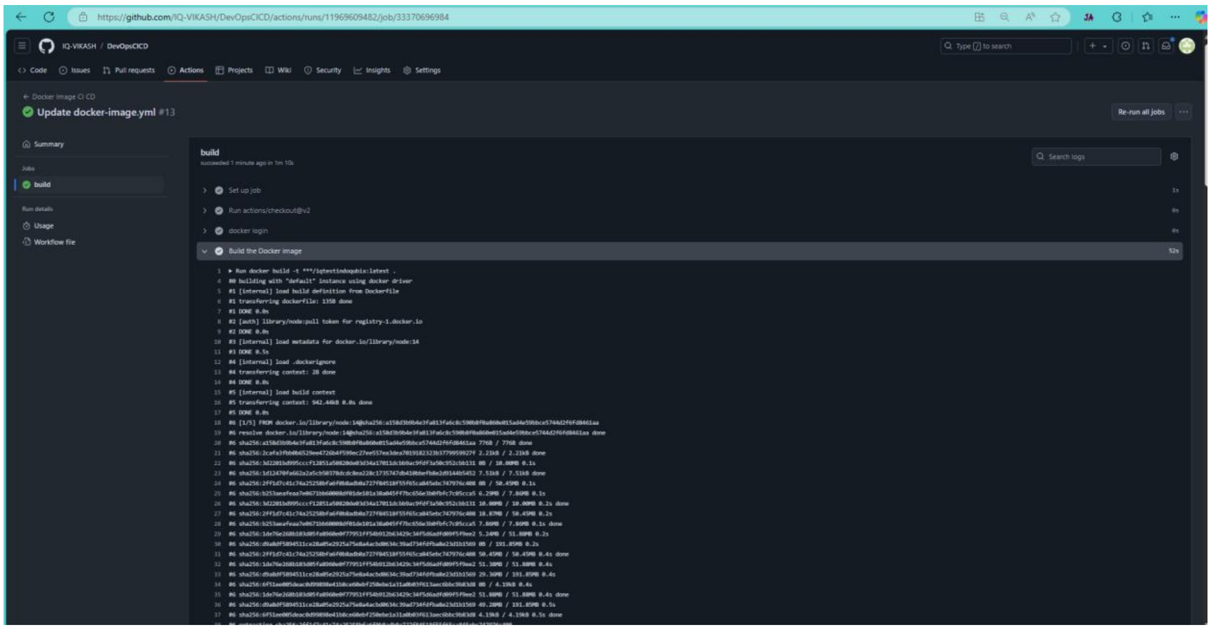
← Docker Image CI CD

Update docker-image.yml #13

Annotations
2 warnings

build
succeeded 2 minutes ago in 1m 10s

```
Set up job                    1s
Run actions/checkout@v2       0s
docker login                  0s
Build the Docker image        52s
Docker Push                   15s
Post Run actions/checkout@v2  0s
Complete job                  0s
```



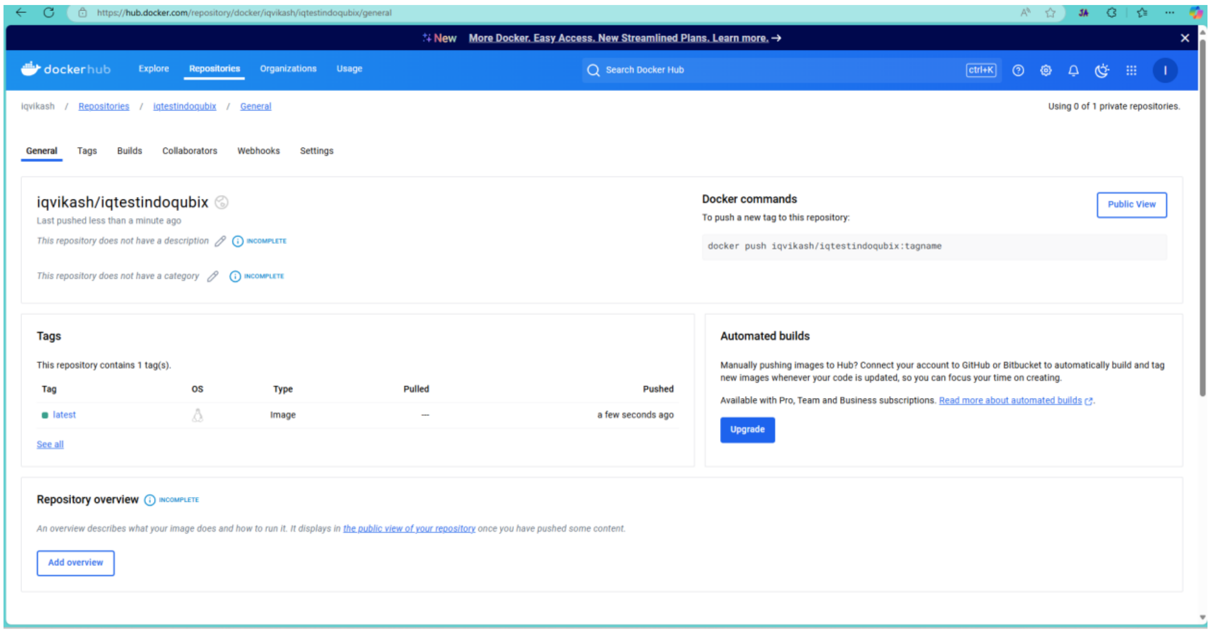
← Docker Image CI CD

Update docker-image.yml #13

build
succeeded 1 minute ago in 1m 10s

```
Set up job                1s
Run actions/checkout@v2   0s
docker login              0s
Build the Docker image    52s
```

```
▶ Run docker build -t ***/iqtestindoqubix:latest .
#0 building with "default" instance using docker driver
#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 135B done
#1 DONE 0.0s
#2 [auth] library/node:pull token for registry-1.docker.io
#2 DONE 0.0s
#3 [internal] load metadata for docker.io/library/node:14
#3 DONE 0.5s
#4 [internal] load .dockerignore
#4 transferring context: 2B done
#4 DONE 0.0s
#5 [internal] load build context
#5 transferring context: 942.44kB 0.0s done
#5 DONE 0.0s
#6 [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b0f0a860e015ad4e59bbce5744d2f6fd8461aa
#6 resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b0f0a860e015ad4e59bbce5744d2f6fd8461aa done
#6 sha256:a158d3b9b4e3fa813fa6c8c590b0f0a860e015ad4e59bbce5744d2f6fd8461aa 776B / 776B done
#6 sha256:2cafa3fbb0b6529ee4726b4f599ec27ee557ea3dea7019182323b3779959927f 2.21kB / 2.21kB done
#6 sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952cbb131 0B / 10.00MB 0.1s
#6 sha256:1d12470fa662a2a5cb50378dcdc8ea228c1735747db410bbefb8e2d9144b5452 7.51kB / 7.51kB done
#6 sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 0B / 50.45MB 0.1s
#6 sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5 6.29MB / 7.86MB 0.1s
#6 sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952cbb131 10.00MB / 10.00MB 0.2s done
#6 sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 10.87MB / 50.45MB 0.2s
#6 sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5 7.86MB / 7.86MB 0.1s done
#6 sha256:1de76e268b183d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2 5.24MB / 51.88MB 0.2s
#6 sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734fdfba8e23d1b1569 0B / 191.85MB 0.2s
#6 sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 50.45MB / 50.45MB 0.4s done
#6 sha256:1de76e268b183d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2 51.38MB / 51.88MB 0.4s
#6 sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734fdfba8e23d1b1569 29.36MB / 191.85MB 0.4s
#6 sha256:6f51ee005deac0d99898e41b8ce60ebf250ebe1a31a0b03f613aec6bbc9b83d8 0B / 4.19kB 0.4s
#6 sha256:1de76e268b183d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2 51.88MB / 51.88MB 0.4s done
#6 sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734fdfba8e23d1b1569 49.28MB / 191.85MB 0.5s
#6 sha256:6f51ee005deac0d99898e41b8ce60ebf250ebe1a31a0b03f613aec6bbc9b83d8 4.19kB / 4.19kB 0.5s done
```


iqvikash / Repositories / iqtestindoqubix / General

Using 0 of 1 private repositories.

iqvikash/iqtestindoqubix

Last pushed less than a minute ago

This repository does not have a description INCOMPLETE

This repository does not have a category INCOMPLETE

Docker commands

To push a new tag to this repository:

```
docker push iqvikash/iqtestindoqubix:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	a few seconds ago

Automated builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

Repository overview INCOMPLETE

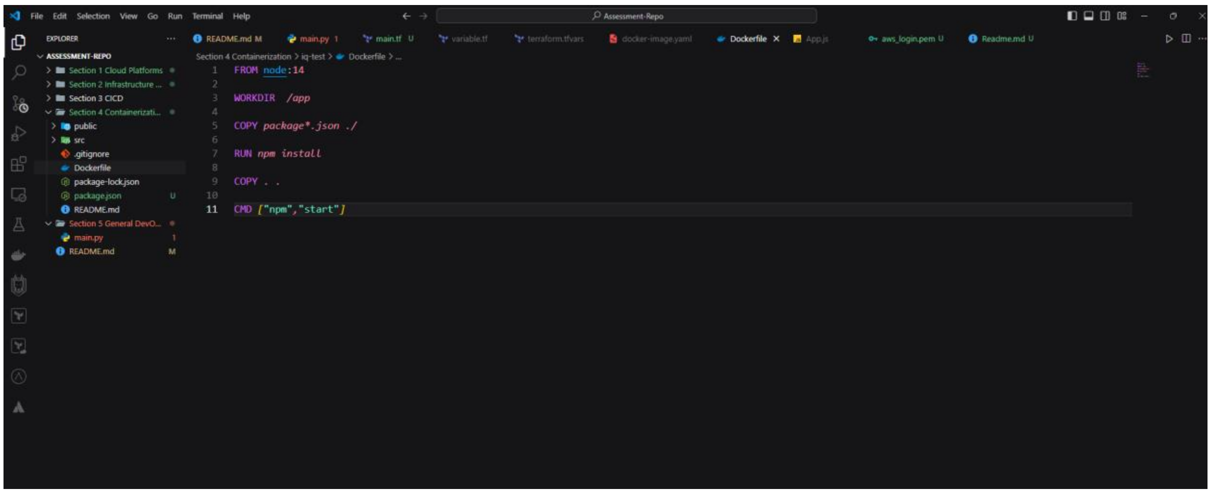
An overview describes what your image does and how to run it. It displays in the public view of your repository once you have pushed some content.

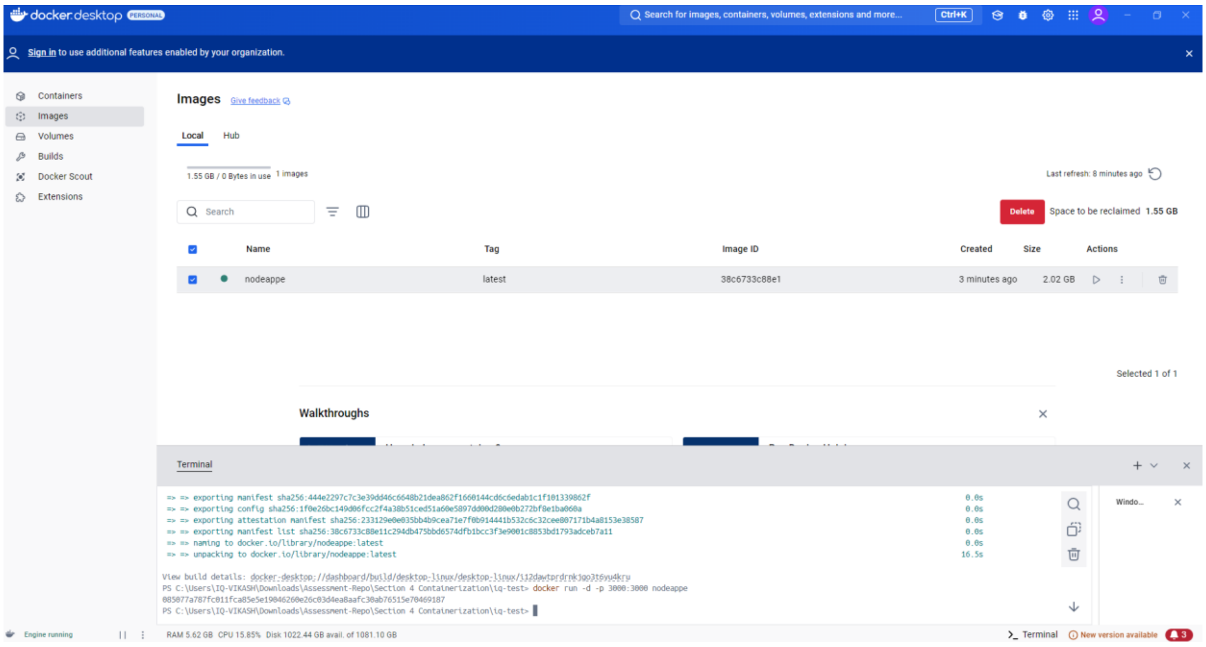
## Section 4: Containerization (60 minutes)

# Write a Docker file for a Node.js application that installs dependencies, copies the application code, and starts the application using npm start.

I have Write the Docker file and copies the application code and start the application using npm start. I have used React App.

Setup Locally my Container is Running.


```
FROM node:14

WORKDIR  /app

COPY package*.json ./

RUN npm install

COPY . .

CMD ["npm","start"]
```


Sign in to use additional features enabled by your organization.

Images

Local | Hub

1.55 GB / 0 Bytes in use 1 images

Last refresh: 8 minutes ago

Space to be reclaimed 1.55 GB

Name	Tag	Image ID	Created	Size
nodeappe	latest	38c6733c88e1	3 minutes ago	2.02 GB

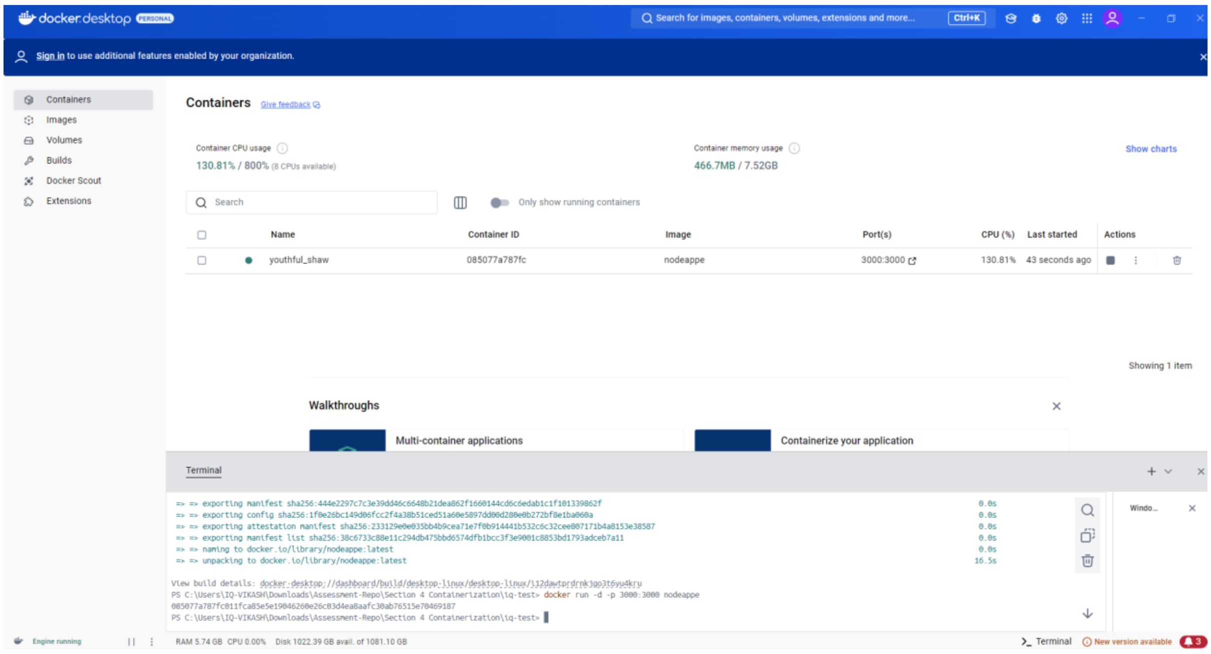
Selected 1 of 1

Walkthroughs

Terminal

```
=> => exporting manifest sha256:444e2297c7c3e39dd46c6648b21dea862f1660144cd6c6edab1c1f101339862f      0.0s
=> => exporting config sha256:1f0e26bc149d06fcc2f4a38b51ced51a60e5897dd00d280e0b272bf8e1ba060a        0.0s
=> => exporting attestation manifest sha256:233129e0e035bb4b9cea71e7f0b914441b532c6c32cee807171b4a8153e38587  0.0s
=> => exporting manifest list sha256:38c6733c88e11c294db475bbd6574dfb1bcc3f3e9001c8853bd1793adceb7a11  0.0s
=> => naming to docker.io/library/nodeappe:latest                                                      0.0s
=> => unpacking to docker.io/library/nodeappe:latest                                                  16.5s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/i12dawtprdrnkjgo3t6yu4kru
PS C:\Users\IQ-VIKASH\Downloads\Assessment-Repo\Section 4 Containerization\iq-test> docker run -d -p 3000:3000 nodeappe
085077a787fc011fca85e5e19046260e26c03d4ea8aafc30ab76515e70469187
PS C:\Users\IQ-VIKASH\Downloads\Assessment-Repo\Section 4 Containerization\iq-test>
```

Engine running RAM 5.62 GB CPU 15.85% Disk 1022.44 GB avail. of 1081.10 GB


Sign in to use additional features enabled by your organization.

Containers

Container CPU usage 130.81% / 800% (8 CPUs available)

Container memory usage 466.7MB / 7.52GB

Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Last started
youthful_shaw	085077a787fc	nodeappe	3000:3000	130.81%	43 seconds ago

Showing 1 item

Walkthroughs

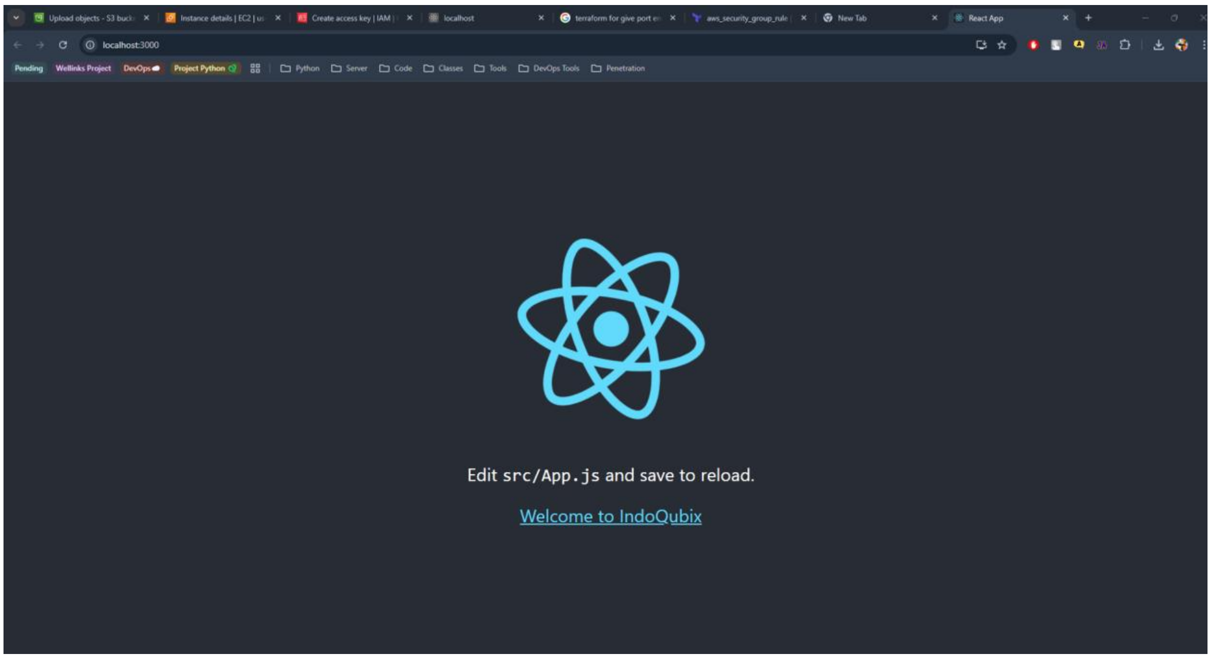
Multi-container applications

Containerize your application

Terminal

```
=> => exporting manifest sha256:444e2297c7c3e39dd46c6648b21dea862f1660144cd6c6edab1c1f101339862f      0.0s
=> => exporting config sha256:1f0e26bc149d06fcc2f4a38b51ced51a60e5897dd00d280e0b272bf8e1ba060a        0.0s
=> => exporting attestation manifest sha256:233129e0e035bb4b9cea71e7f0b914441b532c6c32cee807171b4a8153e38587  0.0s
=> => exporting manifest list sha256:38c6733c88e11c294db475bbd6574dfb1bcc3f3e9001c8853bd1793adceb7a11  0.0s
=> => naming to docker.io/library/nodeappe:latest                                                      0.0s
=> => unpacking to docker.io/library/nodeappe:latest                                                  16.5s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/i12dawtprdrnkjgo3t6yu4kru
PS C:\Users\IQ-VIKASH\Downloads\Assessment-Repo\Section 4 Containerization\iq-test> docker run -d -p 3000:3000 nodeappe
085077a787fc011fca85e5e19046260e26c03d4ea8aafc30ab76515e70469187
PS C:\Users\IQ-VIKASH\Downloads\Assessment-Repo\Section 4 Containerization\iq-test>
```

Engine running RAM 5.74 GB CPU 0.00% Disk 1022.39 GB avail. of 1081.10 GB



Edit `src/App.js` and save to reload.

Welcome to IndoQubix

How to run using These code:-

- Cd iq-test
- docker build -t nodeapp .
- docker run -d -p 3000: nodeapp:latest

# Section 5: General DevOps Practices (30 minutes)

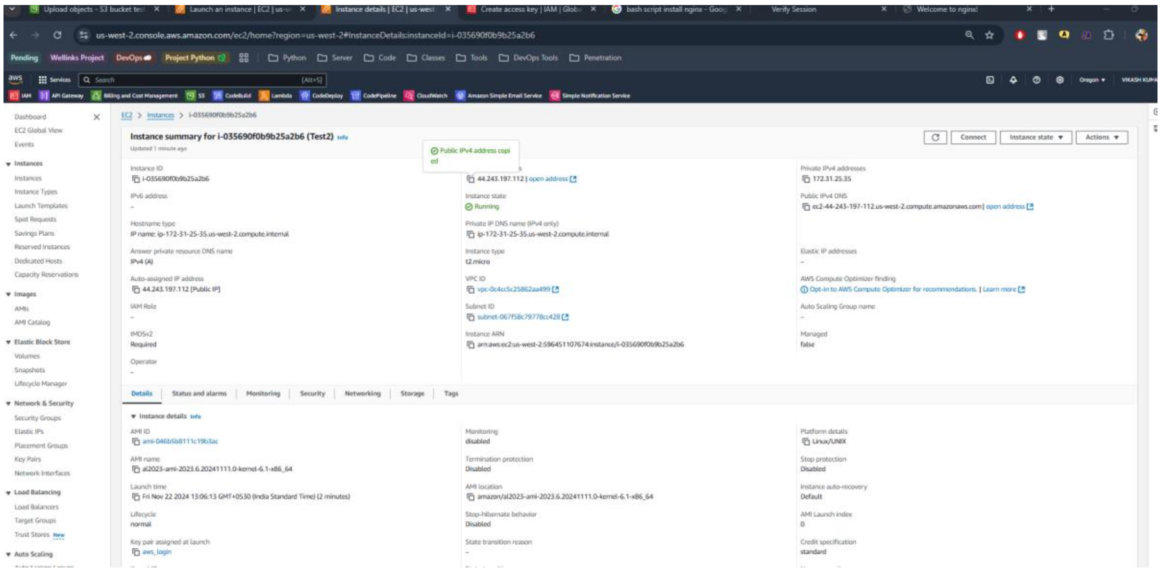
# Theory: Describe a scenario where automation significantly improved the efficiency of a deployment process. Include the tools and techniques used.

We can use Automation for Deploy any application on Server or EC2 Machine

We Can use Python or Bash to Automation.

I have use nginx web server to automation in which using Nginx web server we can sever any file on internet. we can use Automation in different why weather is deployment, sending mail after your application deployed Success or Failed.

In Python we have Boto3 Package in which we can easily automate any AWS tools using this tool and, we can set notification when build failed or deployment Success



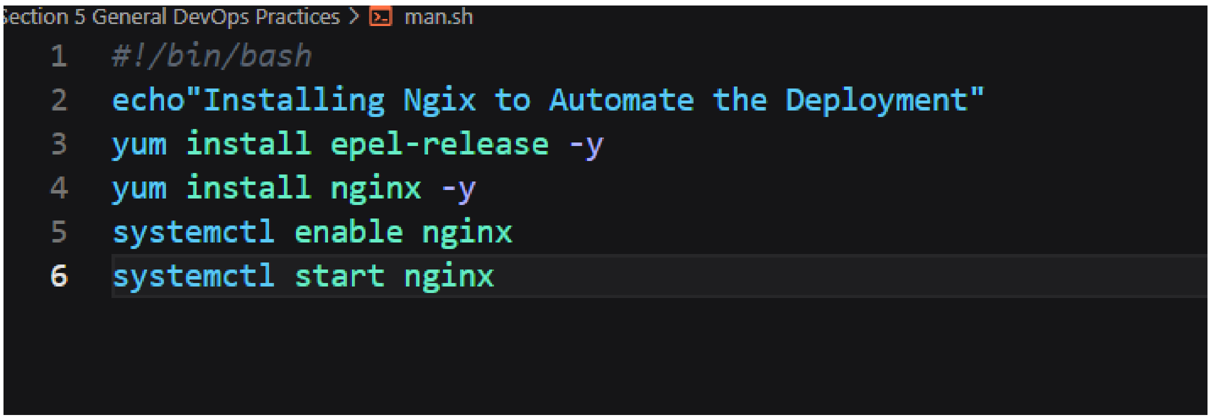
Instance summary for i-035690f0b9b25a2b6 (Test2) Info
Updated 1 minute ago

Public IPv4 address copied

Instance ID i-035690f0b9b25a2b6	44.243.197.112 | open address	Private IPv4 addresses 172.31.25.35
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-44-243-197-112.us-west-2.compute.amazonaws.com | open address
Hostname type IP name: ip-172-31-25-35.us-west-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-25-35.us-west-2.compute.internal	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 44.243.197.112 [Public IP]	VPC ID vpc-0c4cc5c25862aa499	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-067f58c79778cc428	Auto Scaling Group name –
IMDSv2 Required	Instance ARN arn:aws:ec2:us-west-2:596451107674:instance/i-035690f0b9b25a2b6	Managed false
Operator –		

Instance details Info

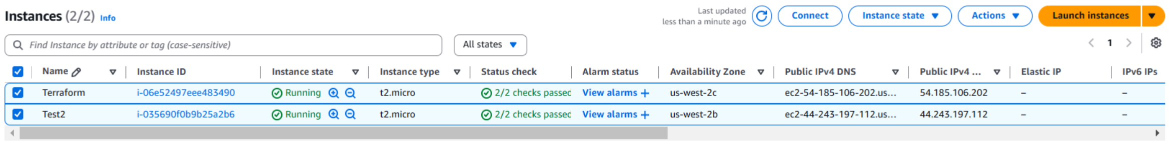
AMI ID ami-046b5b8111c19b3ac	Monitoring disabled	Platform details Linux/UNIX
AMI name al2023-ami-2023.6.20241111.0-kernel-6.1-x86_64	Termination protection Disabled	Stop protection Disabled
Launch time Fri Nov 22 2024 13:06:13 GMT+0530 (India Standard Time) (2 minutes)	AMI location amazon/al2023-ami-2023.6.20241111.0-kernel-6.1-x86_64	Instance auto-recovery Default
Lifecycle normal	Stop-hibernate behavior Disabled	AMI Launch index 0
Key pair assigned at launch aws_login	State transition reason –	Credit specification standard



```bash
#!/bin/bash
echo"Installing Ngix to Automate the Deployment"
yum install epel-release -y
yum install nginx -y
systemctl enable nginx
systemctl start nginx
```

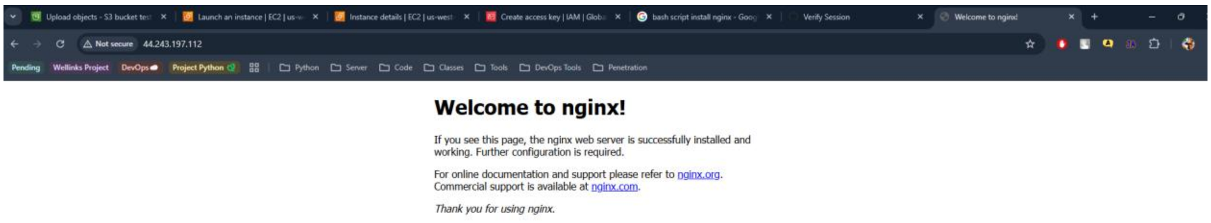
How to Run Above Code-

- Sudo vim main.sh
- Copy these code
- Sudo sh main.sh


Instances (2/2) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs
Terraform	i-06e52497eee483490	Running	t2.micro	2/2 checks passec	View alarms +	us-west-2c	ec2-54-185-106-202.us...	54.185.106.202	–	–
Test2	i-035690f0b9b25a2b6	Running	t2.micro	2/2 checks passec	View alarms +	us-west-2b	ec2-44-243-197-112.us...	44.243.197.112	–	–

ssssss



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Upload objects - S3 bucket test
Launch an instance | EC2 | us-w
Instance details | EC2 | us-west
Create access key | IAM | Globa
bash script install nginx - Goog
Verify Session
Welcome to nginx!
Not secure
44.243.197.112
Pending
Wellinks Project
DevOps
Project Python
Python
Server
Code
Classes
Tools
DevOps Tools
Penetration