

< CFA

Save Q :

LAB : 5

UseCase

1. Database should be kept private
2. Create isolated environment for diff. resources/services
 - It is a secure, isolated private cloud hosted within VPC - Virtual Private Cloud Public cloud
 - You can create your own isolated space / resources. it cannot be accessed outside VPC unless it is peered

NAT gateway : Internet launched in public subnet, and is accessed by private subnet whenever required through NAT gateway

public subnet : has access to internet gateway and internet outside VPC

private subnet - doesn't have access to internet gateway & internet outside VPC

Route Table : controls traffic / determines the path the resource follows to communicate with others.

- If destination resource is within same VPC - local
- If destination resource is outside and it should go through i. internet gateway - public subnet
- ii. NAT gateway - private subnet

Multi AZ - to span resources across multiple availability zone

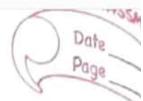
Creating VPC :

In management console

VPC → Create VPC, VPC and more

< CFA

Save Q :



Name tag : myecomvpc , CIDR IPv4 : 10.0.0.0/16

Tenancy : select default

(dedicated - only by you)

No. of availability zones : 2

No. of public, private Subnet : 2, 2

Customize Subnets

public 1 : 10.0.0.0/24

2 : 10.0.2.0/24

private 1 : 10.0.1.0/24

2 : 10.0.3.0/24

NAT gateway : in one AZ

Create VPC

EC2 service

install mysql-client

Launch instance

ecomwebserver

→ network setting

VPC - myecomvpc

subnet - public 1

Auto assign public IP : enable

(whether to create IP address)

security type - ssh

Add security rule : type : HTTP source : 0.0.0.0/0
(Launch)

Launch instance

dBserver

install mysql-server, php-mysql

Network setting :

VPC - myecomvpc

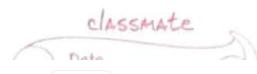
subnet - private 1

Auto assign public IP : disable

Add security rule : type : MySQL Aurora

Source : (ecomwebserver security group rule id)
is only allowed

20/59



Scanned with OKEN Scanner

how to do ssh to database server database server

→ by ecomwebserver

ecomwebserver connect

→ ssh -i labsuser.pem ubuntu@ _____

Public IP of ecomwebserver

to copy labsuser.pem key from local to ecomwebserver

scp -i labsuser.pem labsuser.pem ubuntu@ _____

: /home/ubuntu/. (in local terminal)

before going to dbserver
come to ubuntu@ _____ ↑ copy ecommerce install apache2
and host until before ^
ssh to database server and next connect to dbserver ^

ssh -i labsuser.pem ubuntu@ 10.0.1.24

private IP of dbserver

In dbserver

Create user and database, grant all permission to database.

change bind address, so that it listens from anywhere : cd /etc/mysql/mysql.conf.d

sudo vi mysql.conf

bind address = 0.0.0.0

restart mysql

Come to ecomwebserver

terminal → sudo mysql -u msis -h 10.0.1.24

-P3306 -p

Enter mysql >

Here populate the database :

use ecommerce ;

source onlineshop.sql ;

Connect dB f applicat^n :

cd /var/www/html/ecommence

- sudo vi db.php servername = "127.0.0.1";

username = "msis";

password = "Msis@123";

db = "ecommence";

21/59



UseCase 6

02-09-2023

Change the region to cheaper one

- This can be done by taking snapshot and replicating it into another region

Launch EC2 instance

myinstance

network - default (no vpc)

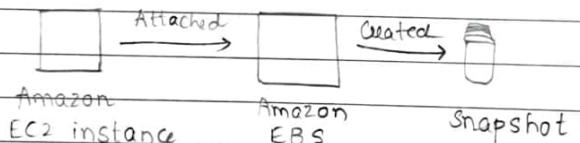
Configure storage : 8 gp2

launch an instance

(In windows : download putty for windows

download ppk key from AWS details

Connection : SSH → Auth → Credentials → Select downloaded Key



06-09-2023

to add additional volume / storage to above instance

Volumes tab in EC2 instances

Create volume

gp2 , size : 1 GB

Availability zone : similar to above zone

Add tag : Name ← key

we can't mount volume if both are in diff. AZ.

My additional Vol ← value

[create]

to use this volume attach to EC2 instance
in Actions → Attach volume.

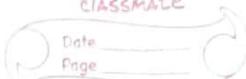
Select instance

Device name : /dev/sdf

Attach volume

22/59

classmate



< CFA

Save Q :

Formatting and mounting : check whether OS detects the newly created volume

in terminal

cd Downloads

ssh -i labsuser.pem ubuntu@ public ip address

to know what diff external and internal storage to system \Rightarrow df -h

to detect additional volume format

to see format of

sudo file -s /dev/n

sudo file -s /dev/xvdf

if it is : data \Rightarrow we have to format it to ext.

to format : sudo mkfs -t ext3 /dev/xvdf
type of format on what storage

sudo file -s /dev/xvdf

df -h still it will be not detected

Now we have to mount it

sudo mkdir /mnt/mydata in mnt directory
create mydata folder
& mount volume
there

sudo mount /dev/xvdf /mnt/mydata
source out

(df -h \Rightarrow /dev/xvdf mounted on /mnt/mydat)

cd /mnt/mydata

create a folder and create a file in that

To take snapshots

Select volume \Rightarrow actions \rightarrow Create snapshot

descriptⁿ : MyVolsnapshot

Tags : key : name

value - my volsnapshot

23/59



CLASS
DnL

< CFA

Save Q :

In Snapshots tab

Select volume → Actions → Create volume

gp2 → 1 → AZ

Tag : key : Name

Value : Restored volume

Attach this volume to EC2 instance

In terminal :

df -h

sudo file -s /dev/xvdf see in attach

volume what is it

We need not format ... we have snapshot of formatted storage to mount

sudo mkfs /mnt/restoreddata

Sudo mount /dev/xvdf /mnt/restoreddata

Check whether folder created in first additional volume is copied in this restored volume

13-9-2023

(Try Elastic Beanstalk)

Select an instance type : based on use case

- General purpose
- Compute optimized
- Memory
- Accelerated computing
- Storage

Instance types : Networking & features
cluster placement group.

interdependent instances has to placed closer to each other or in same server.

3. Specify network settings

4. Attach IAM role (Optional)

owner to speak to

gives temporary privileges to the service (ie to read/accus data) and also what actions can be performed from those services without sharing credentials.

5. User data script (Optional)

(Write shell script to host website

ie ssh, install, git-clone, copy to var/www/
restart apache2.) Assignment

6. Specify storage if we restart instance public

- configure root volume IP address changes, also the server but if we reboot

- attach additional storage volumes. there is no change in storage options:

- Amazon Elastic Block storage (EBS) virtual

- Amazon EC2 Instance Store physical storage of server where instance runs

Other: Amazon EFS, S3

7. Add tags.

8. Security group:

- determines the traffic allowed inside and outside the instance.

9. Identify or create the key pair

remote desktop protocol





We use managed database

25-9-2023

Cloud provider manages everything

You only need to focus on your application
and managing it

You just need to tell cloud provider

what database, version, storage you want.

For example for managed service is S3, RDS

VPC

EC2 - only for webserver

RDS - for databases

Subnet group - group private subnets

Create VPC similar to previous

EC2

Name : mywebserver

Edit n/w settings & choose VPC you created
public subnet.

Similar to previous

Do SSH in terminal

Sudo apt-get update & sudo apt-get install
apache2 libapache2-mod-php php php-mysql
mysql-client.

Sudo git clone https://github.com/sreepathysois/
phpmysql-app.git

phpmysql-app/php/online -

classmate

Date _____
Page _____

sudo cp -rf * /var/www/html/.
sudo chmod -rf index.html
restart apache2

Create managed database server using RDS.

create db.

- standard create

engine type : MySQL Aurora - AWS implementation

version : of MySQL & PostgreSQL

Templates : free tier. only 1 version is created

Settings instance identifier : dbserver

manage master username : admin

master password : Msois1234

He will have all privileges.

instance configuration

burstable

db.t3.micro

storage : type : GP SSD gp2

storage : 20 GB.

VPC : choose your VPC

DB subnet group : Create subnet group

public access : no

VPC security group

Create new name : dbSecurityGroup

unchecked enable backup.

Create

How to connect to this DB server

< CFA

Save



```
cd /var/www/html/database
```

```
mysql -u admin -h
```

endpoint of dbserver

-P 3306 -p

MSois1234

then directly

```
mysql> create database ecommerce;
```

```
use ecommerce;
```

```
> Source onlineshop.sql;
```

```
> exit;
```

```
cd /var/www/html
```

```
sudo vi db.php
```

servername - endpoint

username - admin

pass - MSois1234

db - ecommerce

Restart apache2

Module 8: Database

26-9-23

Section 1: Amazon RDS

managed and unmanaged services *

Challenges of relation database

To overcome these challenges we move to RDS

Managed services responsibilities:

RDS DB instances

Class

Storage

Amazon RDS in a VPC

(the usecase we did above)



28/59



Scanned with OKEN Scanner

Amazon Redshift

27-9-23.

Data warehouse:

Single source of truth to store data of all applications or departments of a organization

To set up a data warehouse in organization we can use redshift : It is fast, fully managed AWS datawarehouse any organization can use to set up cost-effective

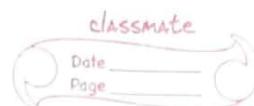
To store peta byte of data and can do analysis on this data using simple queries

1. optimization of queries its advantage is
- 2 it stores data as columns, you can perform easy aggregation, easy compression/encoding so it runs queries in milliseconds (low latency)
- 3 you can parallel queries.

Parallel processing architecture.
leader node : who interacts with client & manage everything, also interact with compute node, it takes source code, divide & give to compute node

compute node : it compiles & runs code given by leader & gives intermediate results then leader node aggregates this intermediate res

32/59



and send it to client and then configure to store either in dynamoDB or S3.

Automation & Scaling

- Manage, Monitor compatibility

Data warehouse:

Single source of truth to store data of all applications of a ~~that~~ organization or departments.

To set up a data warehouse in organization we can use redshift : It is fast, fully managed AWS datawarehouse any organization can use to set up cost-effective

to store peta byte of data and can do analysis on this data using simple ^{SQL} queries

1. optimization of queries its advantage is
- 2 it stores data as columns, you can perform easy aggregation, easy compression/encoding so it runs queries in milliseconds (low latency)
- 3 you can parallel queries.

Parallel processing architecture.
leader node : who interacts with client & many everything. also interact with compute node, it takes source code, divide & give to compute node
compute node : it compiles & runs code given by leader & gives intermediate results then leader node aggregates this intermediate res

and send it to client and then configure to store either in dynamoDB or S3.

Automation & Scaling.

- Manage, Monitor, Scale

Compatibility:

You can use any tools (SQL clients or BI tools) to interact with Redshift.

Use cases:

1. Enterprise data warehouse
2. Big data
3. SaaS.

Amazon Aurora:

It is RDB compatible with MySQL or PostgreSQL.

- It is also managed dB.

It is designed for instant crash recovery if your

- Service benefits: * Primary database becomes read-only

- High availability:

- It stores multiple AZ and later backed up in S3 bucket.

- It

- Resilient design.

(- Redo log files. ^{done by} ← (Crash Recovery of Database)

this file stores insert, update, delete operations made

In Aurora redo log files are not replayed after crash, it does it after every ^{or for crash recovery, read} operations.

EKS - Elastic Kubernetes Service

9-10-2023

Under Compute Category. Container based Services:

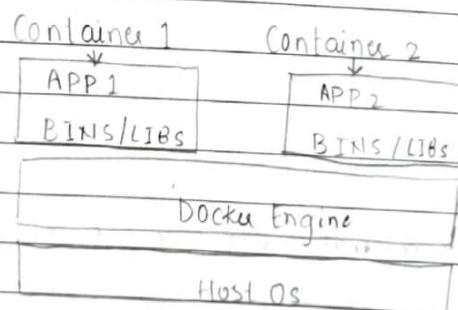
Amazon ECS, EKS, Fargate, ECR.

Why containerization (Virtualization at OS level)

- Makes application hardware independent
- Cross platform dependency
- Interoperability

Software that packages code, optimized OS, packages, libraries into one container.
 (Docker) that are required to run your application.
 (used to containerize application) application.

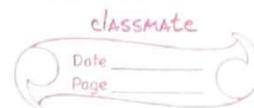
Docker:



Containers are isolated: Any bug in particular container it effects only that container, other containers or host OS are not affected.

Containers are light weight because it has optimized OS (not complete OS) so it loads OS faster.

34/59



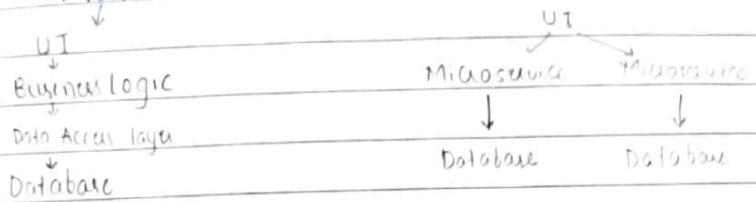
< CFA

Save Q :

In VM's, Memory has to be allocated before it is launched. memory is static whereas in docker, memory need not be allocated, memory grows dynamically.

Web application Architecture.

- Monolithic and Microservice



Monolithic : All features are tied together as single code.

Microservices : ~~all~~ features are treated as diff. module but interact with each other

Advantages of Microservices :

- Extensibility : New features can be added easily
- Error detection & correction is easy, and error in one service doesn't affect other microservices.

Containers are mainly used to run microservices

Docker Hub : Repo that stores images

Components of Docker :

Images : template (Read-only) that contains OS, dependencies required to run application

Dockerfile → Docker Image → Container

35/59

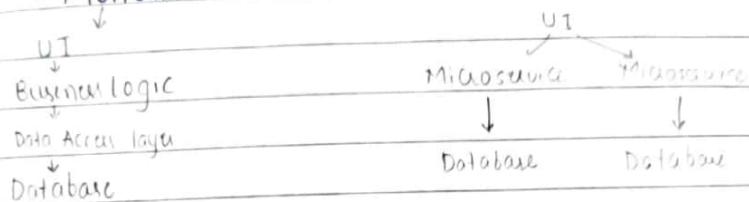
run → start a container

class
Date
Page

In VM's, Memory has to be allocated before it is launched. memory is static whereas in docker, memory need not be allocated, memory grows dynamically.

Web application Architecture.

- Monolithic and Microservice



Monolithic : All features are tied together as single code

Microservices : ~~all~~ features are treated as diff. module but interact with each other

Advantages of Microservices :

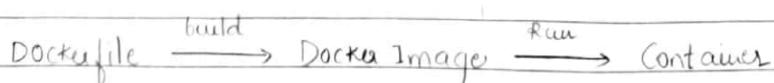
- Extensibility : New features can be added easily
- Error detection & correction is easy, and error in one ^{micro}service doesn't affect other microservices.

Containers are mainly used to run microservices

Docker Hub : Repo that stores images

Components of Docker :

Images : template (Read-only) that contains OS,
 Container : dependencies required to run applic.



Sudo apt-get install docker.io

Sudo systemctl status docker

To test docker is installed & is able to run

Sudo docker run hello-world interacting with
image docker engine through
cli.

do we have any images in local system

Sudo docker images

To pull images

sudo docker pull centos:latest

Start centos and run a container

Sudo docker run --name mycentos -it -d
image name centos:latest contains name

-it → start container interactively and later interacted

-d → run container in background through terminal

If container is started successfully it gives Id

Sudo docker ps → lists all containers that are running

details about containers that ran, exited, running, etc

Sudo docker ps -a

go inside container -it

sudo docker exec ^mycentos bash

root@... # cat /etc/lsb-release through which shell u want to interact

yum update

exit name resolution error

Sudo docker run --name myubuntu -it -d
ubuntu:latest

Sudo docker ps

Sudo docker exec -it myubuntu bash

root@... #

```

cat /etc/os-release
# apt-get update
# apt-get install apache2
# cp /download sample index.html
# service apache2 status
# service apache2 start

# cd /var/www/html
# ls
# rm -f index.html
# vi index.html => vi is not installed so
# apt-get install vim <install>
# vi index.html
copy html code.

```

to create image from running container
in other terminal:

sudo docker ps

sudo docker commit 33... (containerId)

myapache dockerimage:latest < name to your new image

sudo docker images: this should be present

→ sudo docker run --name mywebapp -it -d

-p 3000:80 myapache dockerimage:latest

-p => to expose the port where this apache application can run → remap

• sudo docker ps

In browser 172.3000 it cannot load.

• We have not told image that apache to start.

sudo docker exec -it mywebapp bash

service apache2 status

cd /var/www/html => ls index.html => cat index.html

37/59



class
Date
Page

< CFA

Save Q :



`sudo service apache2 start`

`sudo docker images`

to push to docker hub

`sudo docker tag myapache2 srujan23/`
`image1a` → `myapache2.srujan23/`
`latest`

`sudo docker push srujan23/`

`sudo docker login`

`sudo docker pull`

Building image using Dockerfile.

- 1 `FROM` → installs base OS mentioned
- 2 `RUN` → runs the specified commands inside OS
- 3 `ADD` → Add files from source to destination
or `COPY`
- 4 `ENTRYPOINT` → as soon as container starts
- 5 `CMD` → it is the first task to be done.
eg: `ENTRYPOINT apachectl -D FOREGROUND`
- 6 `ENV` → if your application needs to set up some environment variables
- 7 `EXPOSE` →

create dockerapp folder
& create index.html

Create Dockerfile:

`vi Dockerfile`

→ `FROM ubuntu:latest`

`RUN apt-get update`

RUN apt-get install apache2 vim -y

ADD index.html /var/www/html

(ADD : /var/www/html → copy all files in pwd to /var/www/html)

EXPOSE 80

ENTRYPOINT apachectl -D FOREGROUND

(WORKDIR /var/www/html → cmd's after this should be executed in this directory)

ENV /var/www/html)

To build image:

- sudo docker build -t apachewebapp:v1 =
image name tag name

sudo docker images

- sudo docker run --name apachewebapp -it -d
- p 8081:80 apachewebapp:v1

Dockerize Node.js web application → link

(Only developer side)

sudo mkdir nodejsapp & cd

sudo apt-get install nodejs npm

[package.json → Has all the requirements, dependencies are specified]

(MEAN : MongoDB, Express, Angular, Node.js)

Express is a framework

[To install all dependencies go to path where package.json and run npm install]

" nano "

Sudo vi package.json. copy paste from the link

sudo npm install -y

< CFA

Save



`sudo vi server.js` copy code from link
change const PORT = 8082

`Sudo node server.js`
172.16.51.124:8082

Now dockerize this

`sudo vi Dockerfile` copy code from link
change expose 8082

`Sudo docker build -t nodeapp:v1 .`

`Sudo docker run --name nodeapp -it
-d -p 8082:8082 nodeapp:v1`

11-10-2023

to restart container after reboot/restart

`Sudo docker restart myubuntu` container name

removes all containers in one go (running as well)

`Sudo docker rm -f $(sudo docker ps -a -q)`

Volumes : to store data of container
irrespective of container state.

`Sudo docker run --name myubuntu -it
ubuntu:latest`

`root@... # mkdir /app`

`cd app`

`touch test.txt`

`cat >> test.txt`

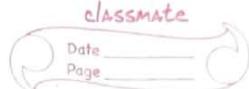
Enter (Hello welcome to docker volume)

`cat test.txt`

`exit`

Thinks that :: of any error or bug
container created/removed

40/59



Scanned with OKEN Scanner

`sudo docker rm -f myubuntu`
 Now we don't have the backup of the data in this container.
 So to persist data we use volumes.

`mkdir dockervolumes`

`cd dockervolumes`

`mkdir app`

`sudo docker run --name myubuntu -it -v`

`app: /app ubuntu:latest`

Some folder on folder in

host data has container

to be stored

(in host volume)

sharing folder
with container

`ls` app folder is present.

`touch test.txt`

`cat >> test.txt`. Enter some data

`cat test.txt`

Now if this container is accidentally removed, the data is stored in shared folder app

`sudo docker rm -f myubuntu`

`sudo docker run -i -v -name ubuntutest -it -v`

`app: /app ubuntu:latest`

→ Here we can get the data that was removed

`cd app` → `ls` → `cat test.txt` → exit.

But this is fine if both have same file system.

If both have different file system, docker came up with docker managed volume : instead of creating volume in host, docker will create volume!

41/59

`sudo docker volume create myvolume`

`sudo docker volume ls` mounted

To check which volume is present in host

`sudo docker inspect myvolume`

`sudo docker volume create myvolume`

`sudo docker volume ls` mounted

To check which volume is present in host

`sudo docker inspect volume myvolume`

`-v`
myvolume
volume managed by docker

`sudo docker run --name ubuntuvolume -it
--mount source= myvolume, target=/app
ubuntu: latest`

`# ls`

`cd app`

`touch file1.txt file2.txt`

`cat >> file1.txt` Enter data into files

`cat >> file2.txt`

`exit.`

`sudo docker run --name centosvolume -it`

`--mount source= myvolume, target=/app
centos: latest`

`# ls`

`cd app -> ls` we get files created in ubuntu

Docker Compose : uses microservice architecture

Eg: For ~~to~~ To split application into multiple containers, tie them together and start, run them together.

Eg ~~host~~ dynamic website. Containerize economics
1 container presentation layer 2. Database.

`sudo git clone https://github.com/sreepathi/phpmysql-app.git`

`cd phpmysql-app /`

`ls`

42/59

classmate

Date _____
Page _____

`vi docker-compose-updated.yml`

→ This file tells that it contains 2 containers

1. web 2. db. ↕ It shows how these containers are started

< CFA

Save



vi docker-compose-updated.yml.

→ This file tells that it contains 2 containers

1. web 2. db. and how these containers are started

cat php/online-shopping-system/Dockerfile-update-webapp.

cat php/online /database/
Dockerfile-db

(docker-entrypoint-initdb.d) where all sql files are stored

→ sudo vi php/
change → FROM php:7.4-apache

To install docker compose:

sudo apt-get install docker-compose

Sudo docker-compose -f docker-compose-updated.yml
up --build -d

bring up all
containers

start in background

(block press deploy with docker-compose (node) link
cd in home)

mkdir wordpress

cd wordpress/

sudo nano docker-compose.yml

change ports to 8009:80.

copy code from
link

Sudo docker-compose up --build -d.

43/59

clan
Date
Page

Kubernetes : (K8s)

16-10-2023

Consider you don't have sufficient resource or network issues on hosts or there is no replica of your container (ie scaling up)

So we need a service to monitor health of this container, scaling, to manage workload. and the service is Kubernetes.

BORG - Container management service

Kubernetes is a light weight

The containers that we have pushed to dockerhub, k8s kubernetes pulls it and deploys it. (through Automation)

Kubernetes Architecture & Components

Master node, Worker node (currently it supports 5000 worker nodes)

Worker node : Runs your container application

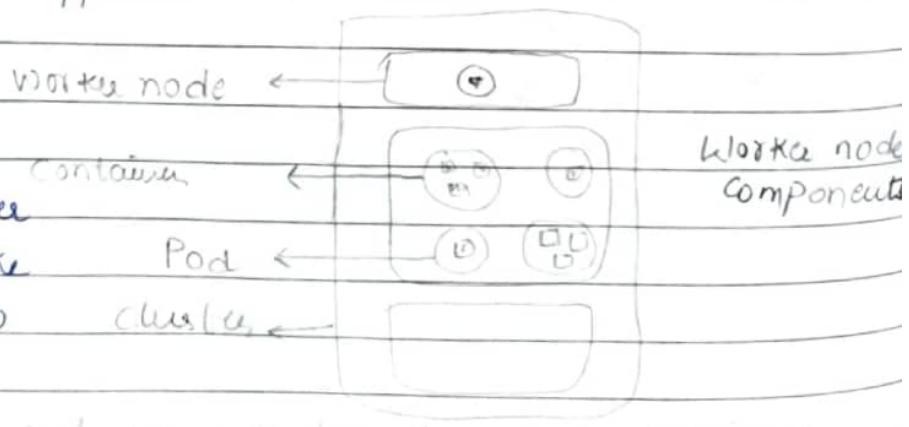
Master : manages everything.

It is to keep containers isolated

→ Pods : Basic scheduling unit of K8s

It is collection of compute resources to run your applications. CPU, SSD, n/w performance

You should
install container
engine to make
worker node to
work



1 pod may contain 1 or more containers.

state of cluster to desired state

Date _____
Page _____

It is where all administrative tasks are performed

Master → Kubernetes (command), UI (K8s dashboard)

API - web interface to interact with master (to do tasks like create, manage, delete, update real K8s Object/pod) ←

API server It does these → Gateway to your complete cluster.

schedules all work in the form of services & pods

Scheduler : schedules your pod by finding best possible worker node as per requirements / specification
It also keeps track of state of all pods

Control Manager : It manages the health of pod or

cluster or node
→ Node, Deployment, Service, Endpoint, Replica Token, Replica
(node) (pods) (end points) (Service is working fine)

subordinates

etcd : distributed key value dB.

(sharding) Complete cluster info is stored here
or current state of cluster.

* Worker node components

Kubelet : primary node agent which has to be installed in all clusters/nodes

- It continuously checks health of node by sending heartbeat to control manager.
- It also checks health of pod
- If node is unhealthy, Kubelet informs control manager.
It restarts by its own if it doesn't start ↑

Kube-proxy : It takes care of networking between hosts, pods, containers.

- It also exposes the services to internet
of container application.

Container
Runtime

< CFA

Save Q :

Kubernetes Pod & Network.

19-10-2023

Adviser: continuously monitors the metrics

Overlay Network: Flannel, Calico, Wanet

Kubernetes Container Deployment.

Load Node balance, Node port, Cluster IP

You don't want to give access to outside world

Deployment is object used to create a pod & run container

Service file:

Demo:

ssh ansible@172.16.51.50
kubernetes master

Get details of cluster:

sudo kubectl get nodes
hostname cat /etc/hosts
Master has role → control-plane

sudo kubectl get pods --all -namespace ^

Create deployment and expose service

\$ sudo kubectl create deployment nginx --image=nginx:latest --replicas=2

To see whether deployment is created

sudo kubectl get deployments

sudo kubectl get pods



< CFA

Save



classmate

Date _____
Page _____

`sudo kubectl expose deployment nginx --port=80`
name of deployment
`--type=NodePort` → automatically remaps port
type of service no bkt 3dt to 65k

`sudo kubectl get svc` service
Go to browser · give IP address of worker node : port

Delete deployment & service

- `sudo kubectl delete deployment nginx`
- `sudo kubectl delete svc nginx`

`sudo kubectl create deployment mybookalbum`

`--image=sredocku123/bookmoviealbum:latest`

`sudo kubectl get pods`

Get details of pod

`sudo kubectl describe pod mybookalbum-` name of pod created

`sudo kubectl expose deployment mybookalbum`

`--port=80 --type=NodePort`

`sudo kubectl get svc`

Go to browser IP address of worker node : port /index.php

Now we can write these command in deployment file & service file

`cd basic-php-website`

`vi bookalbum-deployment.yaml`

`vi -service`

47/59



Sudo kubectl apply -f bookalbum-deployment
Service

sudo kubectl get pods
get svc

kubecfg, Kubeadm, Kubelet

install weavenet

sudo Kubeadm init master

sudo Kubeadm join

In master:

Kubein \$ sudo Kubeadm init

Your control plane has initialized successfully

\$ mkdir -p \$HOME/.kube

sudo cp -i /etc/

Sudo chown

export KUBECONFIG

→ Sudo chown -R msis:msis /etc/

Kubernetes/admin.conf

kubectl apply -f https://

kubectl get nodes → master, node should ready

Kubectl get pods --all-namespaces

here all status should be running

Then you can create deployment and svc



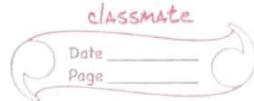
sudo systemctl restart docker

sudo swapoff -a

→ sudo Kubeadm reset

sudo Kubeadm init

48/59



classmate

Date _____
Page _____

28-10-2023

Serverless Computing:

e.g. AWS Lambda.

Under Compute Category

When to opt for this.

- Event triggered or scheduled based.
- When your code needs to run for only some amount of time. (We don't need any servers, computing resources, managing servers.)

① When customer uploads any object to bucket, it should call Lambda and check what type of data is uploaded (.jpg, .png,), and that has to be updated in CloudWatch, and can be displayed through CloudWatch Log.

Cloud Watch monitors

We need to create IAM role so that Lambda can access bucket data and CloudWatch.

read policy

put / post

1. Create a bucket

bucket name : msislamdas3test

us-east-1

Just click default → Create bucket

Go to bucket & upload file.

2. Create IAM role

Policies should \Rightarrow

- i. What are the actions you perform mention
- ii. Whether these actions are allowed on what resource. i.e S3, logs
- iii. or denied

Go to IAM → Policies → Create policy.

JSON → paste code → next

Policy name - s3lambda-read-policy → next

49/59

Date _____
Page _____

next goto roles in IAM only

Create role → AWS service

use case → Lambda → next

Select S3LambdaRolePolicy → next

role name → S3LambdaRole → create.

3. Go to Lambda

Functions.

Create → name: s3lamdatest

Runtime → python 3.9

Architecture → x86_64

change default executive role → use existing role
→ Create

Code → paste code in Lambda function

→ deploy

Add trigger → S3 bucket

bucket → msistamdaS3bucket

Event type → all object create events

✓ I acknowledge → add.

Test → to test your code works properly before releasing

Eventname - mytest
private

Template: S3 put

Event JSON.

input → change region, bucket, name of image
s3 (in ^{down}arn & name) Object(key:
Save & Test

View Cloud Watch Logs

② Create thumbnail of the image you upload in src bucket and put it in dest bucket.
Event trigger function - create thumbnail whenever image is uploaded and push it to dest bucket

Create 2 s3 buckets (src, dest).

1. msisrcbucket name all default create upload an image.
2. misdestbucket name all default create msisrcbucket-resized

Create IAM policies:

IAM → Policies → Create Policy

JSON → paste code → next.

policyname : s3lambdathumbnailpolicy → Create

Go to Roles

Create role → AWS service

use case → Lambda → next

Select the above policy → next

role name → s3lambdathumbnailrole → Create

Go to Lambda → Functions

Create → name : s3lambdathumbnail

runtime → python 3.9

Architecture → x86-64

Executive role → use existing role → above role → Create.

Add trigger → s3 bucket

bucket → src bucket

Event type → all object create events.

BT acknowledge → add.

51/59



< CFA

Save



For write scheduled based trigger to stop & restart CC instance

If you have your own libraries or dependencies for your code to run you have to package it all. (It is lambda function package). Then we use pillow library

To create package - install all dependencies and make zip file of all this along with code also

Code → upload from → select zip folder

Test →

event name - mytestevent

private

template : s3 put

Event JSON

change region, bucket, name of image

Save & test

Syllabus: Database : diff b/w managed & unmanaged
diff b/w sql v/s nosql

RDS, with use case

Dynamo DB use case

Redshift, Aurora

Compute : containers ie Docker → all with a docker file for it

k8s - need of it, Architecture

how you deploy & expose service

Serverless Computing

Module 10:

7-11-2023

Automatic Scaling and Monitoring.

Product Environment

cloud infrastructure for scaling automatically, highly available, fault tolerance, elastic

no scaling quickly.

Theory:

- Elastic Load Balancing
- Amazon Cloud Watch
- Amazon EC2 Auto Scaling

1. Elastic Load Balancing:

You cannot provide multiple users concurrent access with only 1 server. In modern era; we use more servers.

ELB which accepts requests, then divert traffic of client to targets (EC2, containers, IP address, Lambda funct's)

- also divert it to multiple availability zone
- Elastic → scales on its own.

Types of Load balancer

1. Application LB

2. N/w LB : faster compared to application LB
designed to handle more requests :: communication betⁿ machines

3. Classic LB → No target groups.

How ELB works:

Listener : process that checks for connections, request too : what protocol it should listen to
things " port when it receives request on this it directs it to different target.

53/3

Configure Client

Create CloudWatch Metrics etc.

Date
Page

< CFA

Save Q :

→ configure listener.

1 Create target group, register them

Target group : collection of targets.

LB tells listener to send requests to target group.

You can have multiple listeners

Load balancer also checks health of targets
i.e. it sends ping to all targets.

- if it receives response back target is healthy
- if it doesn't receive response successively for 3 times, then target is unhealthy
- & LB doesn't divert request to that target traffic

Load balancer Monitoring:

1. Cloud watch:

continuously monitor services & application on top of services.

2. Access logs: doesn't track all API traffic

3. AWS Cloud trial logs: track all API traffic

2. Amazon Cloud Watch:

- To get insight into your AWS services

- It is monitoring & observability

1. monitor 2. collects metrics 3. Alarms 4. CloudWatch Metrics

Metrics: variables that are used to measure performance.

Eg: CPU utilization, Network I/O throughput, disk utilization

Alarms: you can create alarm for particular metric.

Eg: If CPU utilization of EC2 instances goes higher than 60%, send alarm & tell what should be done.

classmate

Date _____
Page _____

Events:

Eg: When EC2 instance change from running to stop state, (take snapshot) route these events to one or more target function.

namespace: AWS resource on which it should be monitored.

3. Amazon EC2 Auto Scaling: service for achieving application auto scaling as mentioned

- fleet mgmt: it makes sure that ^{desired} min. number of servers are running at any point of time.
- checks health of web servers.

Scaling options:

Manual, scheduled, dynamic or on-demand, predictive (AWS Auto Scaling) ← predictive scaling option

Auto Scaling Group:

Scaling out v/s scaling in.

How EC2 Auto Scaling works:

1. Launch configuration: AMI, instance type, IAM role (role abt how to launch instance), security grp, EBS vol, Auto Scaling group.

VPC & subnets, load balancer.

When - type of scaling

Implementation of dynamic scaling

ELB, EC2 Auto Scaling, Cloud Watch
Auto Scaling group.

55/59

Web instance and database

8-11-2023

< CFA

Save Q :



8-11-2023

Web instance and database
are in private subnet, application load
balanced in public subnet

* Create VPC.

VPC and more

name - ecomvpc

IPv4 CIDR - 10.0.0.0/16

No IPv6

No. of AZ - 2 2 public & private subnets
Customize CIDR

public subnet - 1 : 10.0.0.0/24

2 : 10.0.2.0/24

private subnet - 1 : 10.0.1.0/24

2 : 10.0.3.0/24

NAT gateway : in 1 AZ

VPC endpoints : none

S/IO resource that is used to access resources
sitting outside the VPC

* Create EC2 instances

Name : mywebserver

AMI : ubuntu 22.04

Instance type : t2.micro

Key pair

56/1

CLASSMATE
Date _____
Page _____



Scanned with OKEN Scanner

< CFA

Save Q :

N/W

Select created VPC

Subnet : public 1 us east 1a

Auto assign public IP : enable

Create security grp : type → ssh Anywhere

add rule Http 0.0.0.0

★ Create.

After creating do ssh to this instance.

In terminal

scp -i &Downloads/ -> Html.

ubuntu@^{Keypair}: /home/ubuntu/^{SS of application}

Public IP of EC2

→ \$ cd html.

\$ sudo apt-get update && sudo apt-get install apache

\$ sudo mv html myapp/ rename static files

\$ sudo cp -rf myapp/ /var/www/html/.

\$ cd /var/www/html → ls see myapp

\$ sudo vi /etc/apache2/sites-available/000-default.conf

→ Document Root /var/www/html/myapp

\$ sudo systemctl restart apache2.

★ Create one more EC2

Action → Image + template → Create image

name : ecomami

desc : This is AMI of ecom applicat

Add new tag : Key : Name tag

Value : Myecomami

Create image

Create target group : type : instances

name : myecom-targetgroup

protocol : http:80 where target listen to

IP address : IPv4

57/59



VPC : the vpc you created

Protocol version : HTTP1

Health checks : protocol : http
path : /

next. → create target group

Create load balancer :

Type : application LB

Name : myecomelb

Scheme : internet facing

IP add type : IPv4

VPC - the VPC you created

Mappings : check both AZ & choose public subnet

Security groups :

Allow HTTP from anywhere

Listeners & routing:

Protocol → HTTP : 80 ← Port : select target group you created

Create launch template - details about how to launch instance

Name : myecom.launch.template

Description : template to launch ecom website

AMI : the ami you created.

Instance type : t2.micro

Key pair :

N/w setting

VPC - subnet → private 2

Security group :

Storage :

Resource tag : Name : value : myecom

Create

< CFA

Save Q :

Resource tag : key: name . value : myecomweb
Create.

58/59

classmate

Date _____
Page _____

Actions → Create Auto scaling group
name : ecomasg

Launch template : one you created
next → VPC - one " "

AZ & subnets - select 2 private subnets

next → Load balancing : Attach to existing LB
choose from LB target groups - one you create

Health check - turn on ELB health checks

Grace period : 120 sec

Additional settings : monitoring : check enable
next → Group size & scaling policies

desired capacity → 2 , min capacity → 2
max cap → 4.

Scaling policy : target tracking scaling policy

metric type : Avg CPU uti

Target value : 40

warmup : 60 sec

next → add tag Key : name value : myecomweb
Create.

In browser search with load balancer DNS

try to write on ECR instance , auto scaling

group create one more instance

Test scaling policy →