

# CFA

7-8-2023

Docker

Kubernetes

Automation

Kubernetes Package Manager - Helm

GitOps

DevOps : Automating SDLC

Apache spark - Big Data Architecture

MLOps : pipeline where data go through  
manage, build, deploy.

Modules :

M1 : Cloud computing Overview ↵

M2 : Cloud Economics and billing

M3 : AWS Global Infrastructure Overview ↵

M4 : AWS Cloud Security

M5 : Networking and content delivery

M6 : Compute ↵

M7 : Storage ↵

M8 : Database

M9 : Cloud Architecture

M10. Automatic scaling and monitoring

Module : 1

9-8-2023

- Cloud Computing
- Hardware or Traditional disadvantages
- Software Advantages
- .

Cloud Service Models :

- IaaS, PaaS, SaaS
- XaaS - anything as a service

Each service/deployment models have diff. level of control, flexibility, management

- multi tenancy

Deployment models:

Cloud      Hybrid      Private (on-premises)

CFA LAB 1

11-8-2023

- sbin : holds executables that are executed by administrator
- var : holds log and spool data of complete system
- opt : used for installing third party applicat's which has its own configuration.
- mnt : interact with external devices
- dev : list of all usb, bluetooth devices
- etc : holds folders, subfolders and files reg. to configue. system and software

to change hostname:

- sudo vi /etc/hostname

or

sudo hostname newhostname

- sudo vi /etc/hosts <sup>old</sup>  $\Rightarrow$  edit host name  
then restart  
ip a  $\rightarrow$  ip address

lamp stack - linux apache mysql php  
mean, mern.

- To deploy static website

- sudo apt-get install apache2    downloads .deb  
sudo apt-get update

(sudo rm -rf /etc/apt/sources.list.d/jenkins.list)  
→ to remove repository

Systemctl service

→ utilities used to start, stop, restart, enable, disable a service  
when change is done to particular service

sudo systemctl start apache2

Sudo systemctl status apache2 → to check whether apache2 is running/active

sudo service apache2 status

in browser : http://localhost:80 or

http:// \_\_\_\_\_  
ip address

% Sudo vi /etc/apache2/sites-available/000-default.conf  
in Document Root \_\_\_\_\_ html file path  
var/www/html

download free html5

sudo cp Downloads/html5up-paradigm-shift.zip

src

dstn or  
current  
directory

to extract zip file

- sudo exp mkdir myapp

- sudo cp \_\_\_\_\_.zip myapp/.    copy zip file to  
cd myapp  
myapp folder

- sudo unzip \_\_\_\_\_.zip

- sudo rm -rf \_\_\_\_\_.zip    outside myapp directory

Copy everything from myapp to /var/www/html

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

cd /var/www/html

change DocumentRoot to /var/www/html/myapp  
sudo service apache2 restart  
in browser restart  
(sudo ufw allow port no) if firewall blocks

php based application:

- sudo apt-get install php libapache2-mod-php  
github.com/greepathysois/basic-php-website => copy code file
- sudo git clone https://github.com/greepathysois/basic-php-website
- ls => "basic-php-website" folder is created  
bookalbum
- sudo cp -rf basic-php-website/ /var/www/html  
sudo mv basic-php-website/ bookalbum

sudo vi /etc/apache2/sites-available/000-default.conf  
sudo systemctl restart apache2

if there are multiple index files

sudo vi /etc/apache2/mods-enabled/dir.conf  
→ DirectoryIndex index.php index.html  
change priority

restart

16-8-2023

## Module 1

## Section 2 : Advantages of Cloud Computing

**CapEx** : Capital required to acquire, upgrade & maintain physical assets

1. Trade capital expenditure for variable expense
  2. Massive economies of scale
    - We can get lower price for customers becaz of massive customers.
  3. Stop guessing capacity : ∵ it may lead to  
cloud helps to scale      Overestimated server capacity  
on-demand                    underestimated
  4. Increase speed and agility : in terms of organization we can quickly deploy resources required within mins or by click of 3 buttons. (mins. between wanting & having resources). It reduces time
  5. Stop spending money on running and maintaining data centres.
    - only think about projects
    - need not worry of building, maintaining IT resources
  6. Go global in minutes
    - data centres are set up near to customers which is achieved by AWS, who can quickly launch your applicat so that customers don't feel latency in accessing services

## Section : 3 Introduction to AWS

## What are web services ?

Eg: Going to payment gateway from other applicat<sup>n</sup> & giving result back.

Interface definition file

What is AWS?

\* AWS works together like building blocks

Services covered in this course

- Compute Services
- Storage
- Database

3 ways to interact with AWS

- AWS Management Console
- AWS Command Line Interface (CLI)
- Software Development Kits (SDKs)

## LAB 2

17-8-2023

Deploy dynamic website

Lamp stack : linux apache2 mysql php  
[github.com/sreepathysois/phpmysql-app](https://github.com/sreepathysois/phpmysql-app)

Sudo apt-get install apache2 mysql-server  
mysql-client php libapache2-mod-php php-mysql

sudo git clone https://github.com/sreepathysois/  
phpmysql-app.git  
cd myphpmysql-app

Create folder

sudo mkdir /var/www/html/ecommerce

cd php/online-shopping-system

sudo cp -rf \* /var/www/html/ecommerce/.

```
cd /var/www/html/ecommerce/
sudo vi /etc/apache2/sites-available/000-default.conf
    → Document Root /var/www/html/ecommerce
sudo systemctl restart apache2
```

### setting up database :

sudo systemctl status mysql

create db for ecommerce

create user and password , grant permission to ecommerce db.

- sudo mysql -u root -p ] → to connect mysql db  
use root pass

mysql ➤ show databases ;

- > Create database ecommerce ;

- > Show databases ;

Create user who can access this db from local host

- > CREATE USER 'msis'@'localhost' IDENTIFIED WITH  
mysql\_native\_password BY 'Msis@123';  
remote host or %

Grant permission only to ecommerce db (to all db \* \*)

- > GRANT ALL PRIVILEGES ON ecommerce.\* To  
'msis'@'localhost';

go into db

- > use ecommerce ; database changed

- > show tables ; empty

create tables and populate data nq. for applicat?

- > exit

cd /var/www/html/ecommerce/database

ls

vi Onlineshop.sql

- sudo mysql -u msis -p (Msis@123) ?password

mysql > show databases ;

CLASS  
Date \_\_\_\_\_  
Page \_\_\_\_\_

> use ecommerce;  
> show tables;  
> source onlineshop.sql;  
query in + will be executed  
> show tables;  
> select \* from admin.info; > Select \* from brand  
> exit

Connect application and db

cd /var/www/html/ecommerce

- sudo vi db.php

server name = "localhost";

username = "msis";

password = "Msis@123";

db = "ecommerce";

- sudo systemctl restart apache2

## Module 3 AWS Global Infrastructure Overview

### section 1: AWS Global Infrastructure

- AWS Regions : It is a geographical area

In order to achieve fault tolerance, scalability & stability there are diff. regions set, which are isolated with each other.

- Each region provide full redundancy and connectivity to the network.

- each region has 2 or more availability zones

\* Selecting a Region.

\* Availability Zones  $\Rightarrow$  to make highly available, scalable, fault tolerance

3-4 Availability zone  $\Rightarrow$  Each AZ is fully isolated

each AZ has 3-4 datacenter  $\downarrow$  partitions of AWS infrastructure

each has 100 - 1K servers

- AWS recommends replicating data and resources across availability zones for resiliency

### \* AWS datacentres

We cannot know which data centre our resources are provisioned

- Point of presence - 187
- Edge locations - 176
- Regional edge caches - 11

### AWS Infrastructure Features:

- Elasticity & Scale Scalability
- Fault tolerance
- High availability

Create EC2 instance : Launch instance

Name & tags : lamp

Applicat<sup>n</sup> & OS images : ubuntu

Instance type : t2.micro

key pair login : Create new key pair

keypairname: CDCLabTest , RSA , .pem Create key pair

Network settings : check all

Configure storage : 8GB gp2  $\Rightarrow$  Launch

- cd downloads  $\Rightarrow$  sudo chmod 400 CDCLabTest.pem  
In EC2 instance name of private key

Connect  $\Rightarrow$  ssh copy: ssh -i

ubuntu@ \_\_\_\_ \$ ip a  
exit

LAB 3

23-08-2023

## Bullet Proof Architecture for hosting static website

Object storage : which is accessible through url  
↳ Amazon S3 (Simple Storage Service)  
- only host static website

### Cloud front distribution:

- It distributes the website / content throughout the globe

US-east-1

Select S3 service

Create bucket (with domain name)

Bucket name : msiscdc.com → no 2 buckets

can have same domain name, it should be globally unique. (no capital letters, special characters)

region - US-East

Obj. Ownership

(ACL - Access control list → policy that tells who can view, update, delete data)

ACL disabled

block all public access → uncheck

Bucket versioning → disable

default encryption

⇒ to tell bucket to act as a server

go to properties

- edit static website hosting → enable
- index document → index.html
- error document → optional

⇒ to make objects public so that normal customer can use

- select objects

in action tab → make public using Acc

- to enable ACL  
permissions → Obj ownership (edit) → enable ACL

If availability zones is not there to select, then it is global service.

### Cloud Front Distribution:

- origin domain - it detects
- protocol - http only

origin path - to specify if website is in other folder  
origin shield - No. (DDoS attack)

### default cache behaviour

invalidation ⇒ for everytime given edge location comes to origin server and takes the update and serve the users

- Http & Https
- Get & head
- Default root object → index.html

(Cloud pin) to check from where we got the website

Route 53 service : It translates domain name to their corresponding IP addresses.

- create hosted zone
  - domain name → cdcmcis.com
- create record ← what should be done when clicked
  - alias → cloud front distribution
  - Paste : distribution domain name

for subdomain

Create bucket : name → www.cdcmcis.com

go to bucket → properties → static website hosting → redirect requests for an object

host name  $\Rightarrow$  cdcmsis.com  
protocol - http

### Route 53

- Create record  $\Rightarrow$  [www]. cdcmsis.com

Record type : C Name

value : website link from bucket  
www.cdcmsis.com

## LAB 4 Hosting dynamic website in aws 24-08-2023

Go to EC2 service

Launch instance:

- name & tags (tag is local identifier & it is in webserver key pair value pair form)

- Application & OS images (AMI - Amazon Machine Image)

- ubuntu 22.04

- instance type  $\Rightarrow$  t2.micro, size  
family name, generation

- key pair login : select vockey

$\Rightarrow$  download pem key for ubuntu, ppk for windows

- Network settings :

Security group  $\Rightarrow$  set of rules that determine what kind of traffic allowed into system and out of system

inbound rule  
outbound rules  $\Rightarrow$  it is allowed to anywhere

- allow SSH traffic i.e. 0.0.0.0/0

- allow HTTP

- Configure storage (EBS - Elastic block storage service)  
- 8 - gp2

To instance connect and copy (ssh -i)

In terminal

cd Downloads

⇒ sudo chmod 400 labsuser.pem

⇒ ssh -i labsuser.pem ubuntu@

key

public dns from instance

ubuntu@

⇒ sudo apt-get update

⇒ sudo apt-get install apache2 libapache2-mod-php  
php php-mysql mysql-server mysql-client

⇒ sudo systemctl status apache2

check in browser using public dns

to copy data from local system to cloud securely

in new terminal

⇒ cd /var/www/html → ls

⇒ scp -i /home/msis/Downloads/labsuser.pem -r  
source file ecommerce <sup>key pair</sup> ubuntu@  
to copy :/home/ubuntu/. destination <sup>copy</sup>  
<sup>recently</sup> <sup>copy here</sup>

- again in cloud vm

sudo cp -rf ecommerce /var/www/html/.

⇒ cd /var/www/html → ls & ecommerce )

⇒ sudo vi /etc/apache2/sites-available/000-default.conf  
Document Root /var/www/html ecommerce

⇒ sudo systemctl restart apache2

Setting up database :

⇒ similar to dynamic website hosting.

⇒ This process of hosting website is not good becoz  
it has unmanaged database and both application  
and database is at same place

not managed by cloud provider

## Module : 7 Storage

28-08-2023

- EBS
- S3
- EFS
- S3 Glacier - long time data archiving service

### Section 1 : Elastic Block Storage

- It acts as virtual hard disk of a system
  - Its detachable, durable and persistent block storage for EC2 volumes are replicated within same Availability zone
- (non-volatile)  
It holds data even if you turn off system & turn on after long time

#### AWS Storage Options :

##### Block storage v/s Object storage

- change one block (piece of file) that contains the character
- entire file has to be updated
- If huge data has to be stored

EBS : process data at low latency, high performance

- enables to create individual storage volume and attach them to Amazon EC2 instance
- To replicate volume, Snapshots can be used
- It can be backed up automatically to Amazon S3 through snapshots, can be copied from one region to another

Uses include

- Boot volume (to store OS) and storage for Amazon EC2 instance
- Data storage for File System
- Database
- Enterprise application.

## EBS Volume Types :

- Volumes can be reconfigured without stopping

EC2

(SSD drives is mandatory for Boot volume)

## EBS features

- Snapshots
- Encryption
- Elasticity → increase capacity & change types (i.e. HDD, SSD)

## Volumes, IOPS and pricing

General purpose SSD

Magnetic

Provisioned IOPS SSD

## Data Transfer.

(EC2 → unmanaged service)

## Section 2 : Amazon Simple Storage Service (S3)

(It is abstract service :: we don't have any knowledge of infrastructure or how it works)

- Object level storage.
- to access data globally with end point thus service can be used
- bucket is resource which holds objects
- data is stored as objects in buckets
- designed for 99.9% durability :: data is replicated in 3 availability zones in a particular region
- It is a managed service (i.e. customer need not manage any infrastructure, storage, scaling)
- unlimited storage (single object limited to 5TB)

Data at rest

Data in transit

30-8-2023

Date  
Page

## S3 storage classes

### 1 Standard : 4 9's

- ⇒ offers high availability, durability, performance  
applicat<sup>n</sup> has min redundancy  
downtime      or  
no failure
- ⇒ choose if you have frequently accessing data  
and to process
- ⇒ best fit for designing CDN
- ⇒ has rapid processing time.

### 2 Infrequent Access :

- infrequently access data, 3 9's availability
- offers high durability, low latency, high throughput
- Can be used for long term storage, backups,  
disaster recovery
- low cost, high performance

### 3 One-Zone Infrequent access

- less frequently access, but requires rapid  
processing time
- no need of high durability, even if we loose  
data no worries
- used to store metadata of data
- store only in one-zone

### 4 Intelligent Tiering :

- don't know access pattern of data
- automatically moves data from one storage  
class to another ( i.e. standard or infrequent  
access ) to optimize cost.
- pricing : according to when data is accessed

## 5. Glacier

- low cost, secure, long time data archiving
- no worry of speed

3 types of retrieval methods.

- s3 bucket life cycle management → automatically move data on schedule (one method to upload data to glacier) first 30 days standard  
next 6 months one zone infrequent  
next move to glacier
- other method is directly upload to Glacier

## 6. Glacier Deep Archive

- long term data preservation storing 8-10 years
- compliance or regulatory data to be stored
- Both offers high availability, low cost.  
(Glacier + deep archive)

### S3 bucket URLs

- to deploy static website → virtual hosted style url  
bucket name in region code
- to objects → path-style URL  
region code bucket name

Access data anywhere:

1. Management console
2. AWS command line interface
3. SDKs

### Common Usecases.

- Backup & storage
- application hosting
- Media hosting
- Software delivery

### Storage Pricing:

depends on storage class  
requests, data transfers

## Module 6 : Compute

### Section 1 : AWS Compute Services

- EC2 : resizable compute resource
- EC2 Auto scaling : automatically scale EC2 instances

### Categorizing Compute Services

scalability in terms of instances servers

EC2 : Elastic Compute Cloud

desktop

- replacement for on-premise traditional servers

resources to compute (i.e. require CPU, memory)

### Nine Key decisions to launch EC2

#### 1. Select an AMI (Amazon Machine Image)

- it generally contains OS and few pre-installed softwares required to boot up the instance.
- AMI choices  $\Rightarrow$  Quick start

My AMI

Market place AMI

Community AMI

#### - Create my own AMI

#### 2. Select an instance type

determines : memory (RAM)

Processing power (CPU)

Disk space and disk type (storage)

Network performance

### Instance type categories

#### Instance type naming and size

Eg: t3. Large

t  $\rightarrow$  family name

3  $\rightarrow$  generation number

Large is size

31-8-2023

LAB 5

## UseCase

1. Database should be kept private
2. Create isolated environment for diff. resources/services

## VPC - Virtual Private Cloud

- You can create your own isolated space / resources & it cannot be accessed outside VPC unless it is passed)

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 other

- 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

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

how to do ssh to 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 db

ssh to database server and next connect to dbserver

ssh -i labsuser.pem ubuntu@ 10.0.1.24

In dbserver Private IP of 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

Enters my sql >

Here populate the database

use ecommerce ;

source onlineshop.sql ;

Connect dB f applicatn :

cd /var/www/html/ecommerce

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

username = "msis";

password = "Msis@123";

db = "ecommerce";

restart apache2

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)

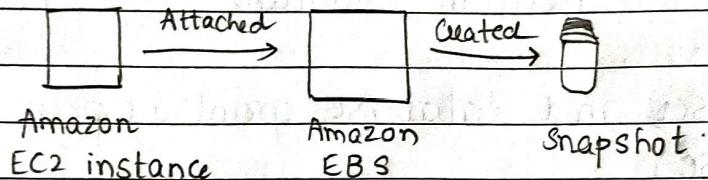
Configure Storage : 8 gp2

Launch an instance

(In windows : download putty for windows)

download ppk key from AWS details

Connection : SSH → Auth → Credentials → Select download 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.

MyadditionalVol ← value

Create

to use this volume attach to EC2 instance

in Actions → Attach volume.

Select instance

Device name : /dev/sdf

Attach volume

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

cd Downloads

ssh -i labscuser.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/x

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

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

cd /mnt/mydata

create a folder and create a file in that

To take snapshots

Select volume  $\Rightarrow$  actions  $\Rightarrow$  Create snapshot -

descript' : MyVolsnapshot

Tags : key : name

value - myvolsnapshot

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/nvdg see in attach  
volume what is it

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

sudo mkfs /mnt/restoreddata

sudo mount /dev/nvdg /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

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

### 3. Specify network settings

#### 4. Attach IAM role (Optional)

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

owner to speak to.

### 5. User data script (Optional)

(Write shell script to host website

i.e ssh, install, gitclone, 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  
storage options:

- Amazon Elastic Block storage (EBS) virtual

- Amazon EC2 Instance Store physical storage of  
Other: Amazon EFS, S3 server where instance launched

### 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