

① Bullet Proof Architecture for hosting a static website
 → It requires ~~ECS~~, S3, CloudFront

1. Origin Server - us-east-1
2. ~~S3~~ → Create S3 bucket → General Configuration
 (Create a bucket name with domain name) (It should be unique)
 → miscde.com (Name) (uncheck)
3. Object Ownership → ACL (disabled) (disable), block all public access
4. Bucket versioning (default) [Uncheck], Tags (leave empty)
5. // To tell bucket to act as a server → Create Bucket
 "html5 up" →
6. Download html page from browser
7. Go inside bucket → click on upload → upload static website code
8. Select Bucket → Properties → static website hosting
 [Defaultly it will be disabled, Enable it]
 → Host a static websites
 Redirect request for an object
 → Index document [Index.html // Name according to source code]
9. Save changes
10. Go to Properties → static website hosting → Endpoint [URL]
 → It gives error so,
11. Go to Objectives → Action tab → Makr public using ACL (enable)
12. In Permissions → scroll → Object Ownership → ACL enable
13. To check → go to Obj → select obj → action → Makr public using ACL
 → I Ack → Save changes
14. Go to CloudFront → Create a CloudFront distribution
 → Origin domain (it lists then select) → Click on give warning
 → click on warning → (Static website endpoint)
 → HTTP only [enable] → HTTP port → 80
15. Origin path (leave it) → Name (will be filled automatically)
 → Add customer header - optional

16. Cache behavior → Yes → Viewer protocol policy HTTP & HTTPS
 → Allowed HTTP method → Restrict view No
17. Web App Firewall Snob Do not enable (choose this)
 HTTP / 2
18. Default root obj
 → index.html (home page) → Create distribution
19. Select security id / ASRL (Copy)
 Then go to browser check the deployment through this "URL"
 & check through the S3 bucket "endpoint"

(2)
LAB-4

Date

Exp No.

Deploying QA type of service for dynamic website.

- " EC2
1. Select EC2 → Launch Instance → Name + tag (dynamic website) (my dynamic service)
 2. App / OS → (Linux) select Ubuntu (64 mb)
 3. Instance type → t2.micro 4. Key pair → select Vokey pair (default)
 4. Network settings → Create security group
 → Security group Allow ssh traffic

 Allow HTTP traffic from internet
 5. Configure storage (default) Select gp2
 6. No. of instance 1 → Launch Instance

Then go back to instance → select then click on connect
 (Check public IPV4 address then)
 → Then in connect, to select (Linux / Ubuntu)
 do ssh via ssh user (Copy)

7. Go to terminal [1] & Downloads , & ls .
8. & sudo chmod 400 labuser.pem [Download labuser.pem for aws launch page]
9. ssh -i labuser.pem ubuntu@ _____ (private) (right side)
10. (ssh is done) & ubuntu@i... & sudo apt-get update
11. & sudo apt-get install apache2 libapache2-mod-php
php php-mysql mysql-server mysql-client -y
12. & sudo systemctl status apache2
13. [New terminal [2]] & Downloads , & cd /var/www/html & ls
14. scp -i /home/hari/Downloads/labuser.pem -t ecommerce/
ubuntu@(part2) :/home/ubuntu/.
15. [back to terminal [1]] & ls ubuntu -- & ls
16. To setup app layer
ubuntu@i... & sudo cp -rf ecommerce /var/www/html/.
17. & cd /var/www/html [ls → ecommerce index.html]
18. & sudo vi /etc/apache2/sites-available/000-default.conf
"change at" Downloads /var/www/html/ecommerce [ls]
19. sudo systemctl restart apache2 [then check ipaddr:localhost]
20. sudo mysql -u root -p [To attach db.] Change page should come
Enter password
- mysql > create user 'hari'@'localhost' identified with
mysql_native_password by 'Msis@123';
- mysql > create database ecommerce;
- mysql > show databases;

mysql > Grant all privilege on ecommerce.* to
'mris'@'localhost';

mysql > use ecommerce;

mysql > show tables;

mysql > source exit

(Terminal 1) \$ mysql @ localhost \$

sudo nano ecommerce/DB.php

Change
serverName : "mris";
username : "mris@localhost";
password : "mris@123";
db : "database";

→ close nano

2) sudo systemctl restart apache2

Then go to browser refresh or type "localhost"

(3) Use case:

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

[It is a secure, isolated private cloud hosted within public cloud.]

⇒ VPC - Virtual Private Cloud

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

It requires - "VPC , EC2 (2)"

① Search → VPC ② Create VPC ③ ④ VPC 4 More (select)

④ Name tag : myecomvpc → IPv6 CIDR block
10.0.0.0/16

→ IPv6 CIDR block ⑤ No IPv6

⑤ Tenancy: Default

→ No. of availability zone: 2 1 2 3 ↗ us-east-1a
 ↗ us-east-1b

→ No. of subnets 0 1 2 (i.e 2)

→ No. of private subnets → 1 0 1 2 4 (i.e 2)

Customize subnets

public subnet 1a 10.0.0.0/24

1b 10.0.2.0/24

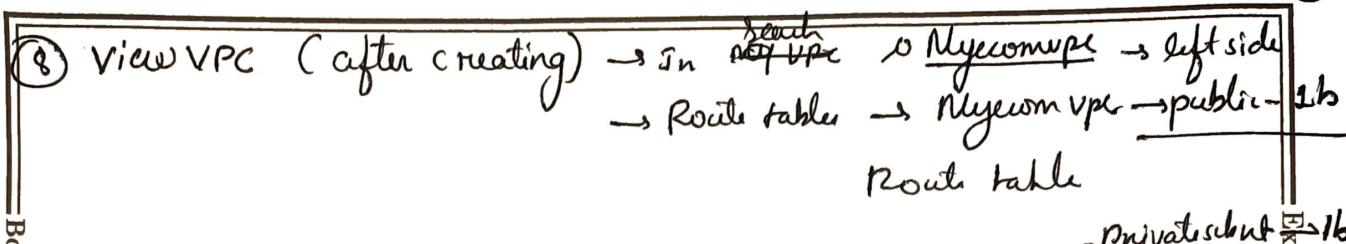
private subnet 1a 10.0.1.0/24

1b 10.0.3.0/24

⑥ NAT Gateway None (i.e 1 AZ)

→ VPC endpoint N/A (None) ⇒ None → DNS op

⑦ → Create VPC



⑨ Launch EC2 instance (Webserver)

- ecomwebserver
- ubuntu
- Instance type : t2.micro
- Key pair → okay
- Network setting : Edit
 - VPC
 - * myecomvpc (Select)
 - Subnet
 - * Public A Z 1a
 - Auto assign IP
 - * Enable
 - firewall (security grp)
 - Create security group
 - SSH
 - Add security group rule
 - (Select) • HTTP → Port 80
 - Source
 - 0.0.0.0/0
 - Configure storage
 - 8 GB
 - Launch

⑩ Launch EC2 (db server)

- db server
- Ubuntu
- t2.micro
- AWS key pair → okay
- New setting : Edit
 - VPC
 - myecomvpc
 - Subnet
 - * private 1 - 1a
 - Auto-assign public IP
 - Disable
 - Firewall (security grp)
 - Add security rule
 - MySQL / Any → Port 3306
 - Source 0.0.0.0/0
 - Configure storage
 - 8 GB
 - Launch

EC2 → webserver → security group → MySQL → ①

(11) Then Go to my instances → select ecomwebserver
[Public IP address is this]

→ ecomwebserver → connect them through SSH (copy)

(12) Download "labuser.pem" from aws console

(13) Open terminal-1 \$ cd Downloads/

→ /D \$ ssh -i part1 → change labuser.pem & remove "quoter"

→ Ubuntu @ ip-10.0.0.54: part1 exit

→ /D \$ scp -i labuser.pem labuser.pem Ubuntu @ ^{addr of webserver}
.com: /home/ubuntu/ _(.com)

[/1 To copy labuser.pem key from local to ecomwebserver]

(14) Open terminal-2 → \$ cd Downloads/ [copy private IP address of
select dbServer instance]

→ /D \$ ssh -i (part1) //ssh to dbserver [copy private address]

→ Ubuntu @ ip-10.0.0.54: ls

→ : ssh -i labuser.pem Ubuntu @ _(part1)

→ Yes

~~→ Ubuntu @ ip-10.0.0.54: sudo apt-get update~~

→ ~~Ubuntu @ ip-10.0.0.54: sudo apt-get install mysql-server php-mysql~~

→ ~~Ubuntu @ ip-10.0.0.54: sudo apt-get update~~ →

→ \$ sudo mysql -u root -p
password: (enter)

mysql > Create database ecommerce;
 mysql > create user 'mis' '%'; identified with
 mysql - native - password by 'Mis@123';
 mysql > Grant all privileges on ecommerce. * To
 'mis' '@' '%'; ~~use~~
 > exit

10.0.1.224

→ Ubuntu @ ip -- : ip - a

→ — \$: sudo apt-get install apache2 libapache2-mod-php
 php php-mysql mysql-client / mysql-server not required

→ ^{vo:} \$: sudo git clone https://github.com/sreepathy901/
 phpmysql-app.git

→ — \$: sudo mkdir /var/www/html/ecommerce

→ — \$: cd phpmysql-app/php/Online-shopping/

→ — phpmysql-app/php/onlineShoppingSystem \$ sudo cp
 sudo cp -rf * /var/www/html/ecommerce.

→ — .system \$ cd ~~/~~ /var/www/html/ecommerce/

→ — ecommerce \$ ls

→ — www \$ cd database/

→ — www/database \$ ls

→ — www/database \$ sudo mysql - u mis ^{host not local} - h 10.0.1.224
 -p 3306 -p ⁵⁴
 db name

enter password : Mis@123

mysql > show databases;
 mysql > use ecommerce;
 mysql > show tables;
 empty set
 mysql > source onlineshop.sql;
 mysql > show tables;
 mysql > select * from admin_info;
 mysql > select * from brands;
 mysql > exit

\rightarrow \$ /var/www/html/ecommence/database & cd ..)

\$ or

webserver

_____, \$ cd /var/www/html/ecommence/database

/clothes: \$ ls

\$ sudo nano db.php

→ change

Servername : " _____ " ^{private IP}

password : " Msis @123 "

username : " msis "

db : " ecommerce "

\$ sudo systemctl apache2

(A) Use case: This lab is designed to show you how to create the Amazon EBS Volume. After you create the volume, you will attach the volume to an Amazon EC2 instance, configure the instance to use a virtual disk, create a snapshot & then restore from the snapshot.

(B)

Change the Region to cheaper one

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

The ~~first~~ service → "EC2, EBS Volume of EC2"

① Launch EC2

- (name myserver)
 - ubuntu
 - t2.micro
 - No setting (default)
 - default VPC
 - security group
 - HTTP
 - gp2
 - Launch

[Note: Note region]

us-east-1

② To add additional volume/storage to above instance

- To add another volume *left side*
- Under EC2 instance select → scroll down
- Volumes
- Create Volumes
- Volume type: General purpose SSD
 - Size(GB) → 16GB → 1Gb, opt
 - "Select availability zone" when instance is shared
 - Select az as in ec2 i.e. us-east-1
 - Snapshot ID (leave as it is)
 - encryption (Uncheck)
 - Add → key → Name
 - Value → Myadditionalvolume
- Create volume

② To use this volume, attach to EC2 instance

→ Volume → Select : Action → attach volume

→ Volume ID (copy)

→ Availability zone → us-east-1

→ Instance → (choose) myserver

→ Device Name : ^{IMP} /dev/sdf (Make a note)

Make a note on /dev/xvdp

→ Check "in-use"

→ Now see if it is in-use or not [Addnd volume]

③ Formating and mounting : [before this copy is add of public key]

Go to terminal

[Download labuser.pem from
aws console]

→ \$ cd Downloads/

→ ~~Downloads~~ \$ ssh -i labuser.pem ubuntu@ public ip address ^{dns}

→ — \$ sudo chmod 400 labuser.pem

→ — \$ ssh -i labuser.pem ubuntu@ —

→ ubuntu@ip — \$ df -h // To know diff btw ext & internal storage

~~df~~ To detect additional volume format +
to see format of volume

→ u — \$ sudo file -s /dev/x

→ u — \$ sudo file -s /dev/xvdf

? /dev/xvdf : data // Means not formatted or mounted
we have to format it to ext

→ u — \$ ext base

→ \$ sudo mkfs -t ext3 ~~file~~ /dev/xvdf \$
Sudo type of format on what storage

→ ——— sudo file -s /dev/xwdf
→ ~~45~~ (diff-h still it will be not detected

Now we have to mount it

$\rightarrow -n - \$ \text{ df } - h$

$\rightarrow u$ In this mount we mount to extend

→ cu : \$ sudo mkdir /mnt/mydata // create a folder

$\rightarrow u \quad s \quad df-h$
(+ nasal /dvn/ xvdf will be attached)

→ cahn — \$ cd /mnt/mydata/

→ mydate & sudo mkfs.ext4 volume data

→ -n- \$ cd volume date

→ a volume data / \$ sudo touch file.txt

→ _____ \$ sudo nano file.txt

_____ \$ 5000.00)
Add date 7/2 Hello welcome to EBS Volume
1 issue 121

This is a date of additional

→ _____ voln & cat file.txt
④ Go to aws lab (To do snapshot)

→ EC2 instance → Go to volume → select mounted volume →

2. Click on IP: My additional volume → Actions → Create snapshot

→ description: Myvolume snapshot

Tag : key value
.....

Name My Volume snapshot

Create

→ (left side in EC2) EBS → select snapshot

→ To use snapshot

→ Create EBS (OS)

→ another volume

→ Select My volume snapshot → Action

→ Create volume from snapshot

→ Volume type

gp2 = 1B → AZ

Tag	Key	Value
Name		RestoredVolume

→ Create volume

Volume → Restored Volume → attach volume

(as in -me) → It comes volume then Note

③ Go to terminal;

→ Ubuntu & -> volume \$ cd

→ Ubuntu & -- 2 \$ df -h

→ -- \$ sudo file -s /dev/x

→ -- \$ sudo file -s /dev/xvdcg see in attach
volume what is it

We need not format ; we have snapshot of formatted storage

to mount;

→ \$ sudo mkdir /mnt/restoreddata

→ \$ sudo mount /dev/xvdcg /mnt/restoreddata
\$ df -h

Session

then see /xvf & xx vcf also added

- → \$ cd /mnt/restoredata/
- 1/ To validate snapshot
- → → \$ /mnt/rebreat : \$ ls file.txt
- → → tar \$ cat file.txt

Gives Hello —

Lab 7 Question:

Cloud providers "manage database" then it manages everything but we have to specify the spec"

①

- Cloud providers manage everything, we only need to focus on your appⁿ & manage it
- You must need to tell cloud provider what db version, storage you want (Eg: for managed S3, RDS)
- "Requirement - VPC, EC2 (only for webserver), RDS (for db)"

① Create VPC (as as previous) [myvpc]

② Launch EC2 instance

→ name: mywebserver, ubuntu, t2.micro, key → Vicky

→ New setting → edit choose [myvpc] [I created vpc]

• public subnet → 1a

→ Security group [Add]
ssh - anywhere

• Enable

→ Gray8

→ Launch

- ③ Go to github copy <http://github.com/sneopathyos/phpmysql-app.git>
- ④ Go to instance → Connect → SSH
Copy ssh -i
- ⑤ Go to Terminal:
- ```

→ $ cd Downloads/
→ Downloads $ ssh -i labuser ubuntu@ -- part
→ $ sudo chmod 400 labuser.pem
→ $ ssh -i labuser -- (path)
→ ubuntu @ in- . . . $ sudo apt-get update & sudo apt-get
install apache2 libapache2-mod-php php
php-mysql mysql-client
→ u -- $ sudo git clone https://github.com/sneopathyos/
phpmysql-app.git
→ u -- $ cd phpmysql-app/php/online-shopping-system/
→ u -- -system/$ ls
→ -- m/$ sudo cp -rf * /var/www/html/.
→ $ cd /var/www/html
→ -- httd $ ls
→ $ sudo rm -rf index.html
C: it shows error, it has two html
$ sudo systemctl restart apache2

```

⑥ Go to aws : Search Amazon RDS

→ [Create db]

① Standard create ○ — // No need to explore

→ Engine options ○ Amazon (MySQL Compatible)

② MySQL (Select)

→ Engine version:

MySQL 8.0.34 (Latest Version)

Availability & → Templates ○ production  
durability

○ Day/Year

○ Free tier (Select)

→ Settings

→ Instance identifier : db server

Setting

master username : admin

master password : Msis1234

// he will have all privileges

→ Instance Config

○

○

② Burstable class (Select)

→ db.t3.micro (Instance type)

Storage : type

→ general purpose SSD (gp2)

→ Allocated storage

Storage : 20 GB

Connectivity : ① Don't connect to an EC2 instance (Select)

○

Page No. : 103

Date: 7/7/2023

Exp. No. :

- Virtual private cloud (VPC)
  - myvpc
- DB subnet group
  - ecomdb group
- Public access
  - Yes
  - No (select)
- VPC security group
  - Create new
- New VPC security group name  
db Security Group

- ← VPC
- P 2
- Database Authentication
    - Password
    - Monitoring
  - Enable (Select / enable)
  - Add new config
  - Database options
  - Backup (unchecked)
  - Enabled automated backup
    - as db stops it starts
    - backup & store
  - Encryption
    -
  - // Remaining same
  - [Create database]

Left side:

Database → select 4!

- Connectivity & security
- Security group
- Inbound group → Edit inbound rule
  - 0.0.0.0/0

[Security grp of EC2 + port but now 0.0.0.0/0 same rule]

- Type MySQL / Aurora
- add custom TCP Port 3306
- // This is only way to connect to mysql client
- Port → 3306

⑧ Go to terminal:

→ ubuntu → /var/www/html \$ cd database/  
 → u → database \$ ls  
 → u → database \$ mysql -u admin -h dbserver endpoint  
 → u → c47heyulya.us-east-1.rels.amazonaws.com:3306 -p 3306 -p  
 → u → password end part → try to remove

Enter password: Msis1234

mysql > create database ecommerce;

mysql > show databases;

> use ecommerce;

> show tables; //empty set

> source Online\_shop.sql;

> show tables

> select \* from brands;

mysql > exit;

→ u → database \$ cd ..

→ u → /var/www/html \$ sudo nano db.php

Server: "db.Sec413...amazon.com" // endpoint

username: "admin"

password: "Msis1234"

db: "ecommerce"

→ u → /html \$ sudo systemctl restart apache2

⑨ Then go to browser → localhost

Exp. No. :

Date :

Page No. : 102

# Auto Scaling

109

## ⑥ Usecase:

- App<sup>n</sup> availability Based on customer demand, Manual, schedule
- Fault scaling, on-demand, several scaling option, dynamic
- Load balancing

## ⑦

App<sup>n</sup> achieves autoscaling. Web instance and database are in private subnet, app<sup>n</sup> load balances in public subnet.

Hosting website requires isolated environment, scale on its own,

→ highly scalable → Elastic → scale on its own

[Soln: Load balancer → Auto scaling → DB on Managed/unmanaged]

→ AMI → VPC → EC2 → Take snapshot of AMI

→ Autoscaling]

→ " Requirements → VPC, EC2

## ① Create VPC:

→ myvpc → VPC setting ② VPC & more

→ naming ecomvpc → IPv4 block 10.0.0.0/16

- IPv4 (default) - Tenancy (default)

- No. of AZ 

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|

 (2)

- No. of Public subnets 

|   |   |
|---|---|
| 0 | 2 |
|---|---|

 (2)

- No. of Private subnets 

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
|---|---|---|

 (2)  
• Customize

10.0.0.0/24 (public 1a)

10.0.2.0/24 (pu 1b)

10.0.1.0/24 (private 1a)

10.0.3.0/24 (pn 1b)

→ NAT Gateway 

|   |   |   |   |
|---|---|---|---|
| N | T | A | Z |
|---|---|---|---|

 (select 3 AZ)

→ VPC endpoint 

|   |   |   |   |
|---|---|---|---|
| N | O | N | S |
|---|---|---|---|

 (select None)

If Remaining default →

Create

① EC2 Instance

→ Name : mywebserver → ubuntu → t2.micro ~~key pair~~ V.

\* → Key pair // Create mykey  
\* Alias Mykey

→ New setting edit → VPC & myVpc

\* → public subnet us-east-1a

Enable

→ Create Security group (anywhere)  
add HTTP → 0.0.0.0/0

1 default → Launch instance

② Go to instance i.e. mywebserver → select & Connect → SSH (copy)

④ Download html sup - paradigm - shift

⑤ Go to terminal;

\$ cd \$ Downloads

→ Downloads \$ ssh -i mykey (path) \_\_\_\_\_ . ~~pass~~

yes

→ Download chmod 400. mykey.pem

(path)

→ D \_\_\_\_\_ \$ ssh -i mykey.pem user @ \_\_\_\_\_ .

→ user @ \$ exit

→ Downloads @ \$ cd

→ scp -i Downloads / mykey.pem (path to some code of ④)  
ubuntu@ \_\_\_\_\_ : /home/ubuntu .

⑥ Go to AWS : EC2 → instance → Action (top left) → Image & template

→ Image name : elomani

→ Cd

→ \$ cd Downloads /  
 → Downloads \$ ssh -i (path) of file cont with mykey.pem user  
 → \$ ls  
 → cd html sup-paradigm shift  
 → shift \$ ls  
 → \_\_\_\_\_ \$ sudo apt-get update & & sudo apt-get install apache2 -y  
 → \$ cd  
 → \$ sudo mv htmlsup /myapp /  
 → To rename to myapp  
 → \$ ls  
 → sudo cp -rf myapp /var/www/html/  
 → cd /var/www/html  
 → html \$ ls  
 → \_\_\_\_\_ html \$ sudo gedit /etc/apache2/sites-available/000-default.conf  
 → Do change  
 Change Document Root to /var/www/html/myapp  
 → \$ sudo systemctl restart apache2  
 Go to browser public ip address

### ⑥ Create AMI

(left top)

- Go to AWS → Search EC2 → Instance → Action → Image & template
- Image name: econmani
- clue: This is AMI of econ appn (description)  
(default)
- Add a tag
 

|         |           |
|---------|-----------|
| Key     | Value     |
| NameTag | myeconami |

 → **Create**

### ⑦ To create target group

- Select → myeconami  → (scroll down lefside)
  - Load balancer
  - Load balancers
  - Target group → **Create target group**

- Change a target type

① Instances (Select)

- name: ~~Target~~ myecomtargetgroup

Protocol port → Http : 80

IP address type: ① IPV4 (selected)

VPC

- VPC: Select the VPC created

→ Protocol Version: ① HTTP Version /

- Health checks

→ Health check path:  Health check

→ Health check protocol:  HTTP

Default

**NEXT**

*(Not mandatory part)*

→ Register target

Available instance

(Don't do anything) (i.e. default)

→ **Create target**

Signature of the Lecturer

## ② ~~left side~~ Create Load Balancing

(left side → load balancing → load balancer)  
Action

(Type) App<sup>n</sup> load balancer | |

**Create** :: It is HTTP

→ load balancer name

myelb

→ Scheme : ① Internet IP address type : ② IPv4

→ VPC (The VPC created is myvpc)

X/w Mapping :

VPC

\* Mappings : Check both AZ, choose public subnet

i.e  us-east-1a (public subnet 1a) Select

us-east-1b (public subnet 1b) Select

→ Security group (Select created security group)

→  Allow http from anywhere

→ MYELB61 which has permission to access anywhere

→ Listener & Routing

▼ Listener HTTP-40

→ Protocol Port Default action info (Select target group)  
HTTP 80  mytargetgroup ▼  
(default)

**Create**

## ⑨ Create Auto scaling group (i.e. Create LaunchTemplate)

In EC2 → Instances left side (scroll down)

→ Launch templates → Create launch template

→ my name: myecomlaunchtemplate

→ Description → Template to launch a ecom website

▼ App & OS Images (AMIs)

AMIs → myAMI (select AMI you created)

→  (Select)

(~~8~~) → ecomami

→ Instance type: t2.micro, key pair: myec2 (selected)

→ Network settings

→ My VPC → private us-east-1a (Private subnet)  
private us-east-1b

→ Firewall

① Select creating security group

→ Common Security Group ▼

MYELB&

→ Storage EBS (default)

→ Resource tag key value (Review key)  
Name: myecomweb  
(Default)

Create launch template

⑩ Go to launch template ① select Action → Create auto scaling group

Go to auto scaling (from left side before instances)

Auto scaling group name: ~~ecom~~ ecomasy

Signature of the Lecturer

→ launch template : ~~choose launch type~~ → **NEXT**

→ Choose Instance

→ New Subnet

→ Select new type (created VPC) (Select private subnet)

A2 →  us-east 1 a private subnet 1

us-east 1 b private subnet 2

→ **NEXT**

→ Configure advanced option select

Load Balancing :

Attach to existing LB

→  Choose from LB target group (Choose or you create)  
Myewtargetgroup  (Selected)

→ VPC endpoint Integration  
default

→  Health checks - turn on ELB health checks  
grace period : 120 seconds

\* → Additional Settings → Monitoring →

Enable group metric collection with cloud watch

**Next**

→ Configure group size & scaling policy

→ Group size : Desired Capacity → 2 , Minimum Cap → 2  
Max Capacity → 4

→ Scaling policy : on-demand

Target tracking scaling policy

→ metric typology : Average CPU utilization

11 (Memory disk) Target Value : 40 2 minute warm up : 60 second

Signature of the Lecturer

- Instance scale in (If Do you want) add → **NGXT**
- Add key : key      Value  
Name      myecomweb → **NGXT** → **Create**

// Then after this go to EC2 instances after launching them,  
2 new EC2 instances will be added

- To check LB is active or not ,  
scroll down to see DNS  
Name      Select **IP** (Copy)
- Go to browser : (part)  
// To see fleet manager  
To validate auto scaling