→ Select EC2 in console

→ Select Region - US East (N. Virginia)

→ Launch an instance

Name and tags : cafewebserver — template for system creating

→ Application & OS Image : — based on region the virtual server
(Amazon Machine Image) runs system ubuntu/redhat/centos

Quick start : ubuntu.

ubuntu 22.04 LTS ami

→ Instance type — hardware necessary to config project
t2 - generation - choose highest
"t" - general purpose instance type.

[t2.micro] → amount of hardware

→ Key pair — different for diff OS.

vockey.

→ Network settings

default one VPC is created in every region.

Space created on public space (VPC) — accessable over internet.

VPC — default.

→ Security group

has a firewall which filters out the requests coming.

Inbound security group

type : ssh
Source type : anywhere
(0.0.0.0/0)

} allows (traffic) request of ssh type from anywhere in the world.

⇒ Configure storage — harddisk

1 x [8] GB general purpose 2

LAUNCH an instance

→ Instances → cafeserver → meta data.

→ Lab → AWS details — PPK : windows

→ SSH → download PEM

→ Copy public EC2

→ cd Downloads → ls → (downloaded.pem)

→ sudo chmod 600 labuser.pem : file permission to be able to read
(same file name.)

→ sudo ssh -i labuser.pem ubuntu@ipaddress
↳ public ip address in meta data

→ ip a.

→ sudo apt-get update -y

→ sudo apt-get apache2..... ← 2nd command in doc file. (deploy dynamic web)

→ sudo systemctl status apache2

→ Browser → http://publicIpaddress from metadata

→ Security group – launch wizard 3

→ Inbound rule – add rule – type HTTP – port 80

→ Source type – anywhere IPv4.
denies every other request type (SSH then HTTP denied add HTTP under security)

→ sudo clone .... [command in doc]

→ sudo mkdir /var/www/html/cafeapp → cd cafe-dynamic-web

→ Copy source code of repo to above created folder
sudo cp -rf mompopcafe/* /var/www/html/cafeapp

→ sudo nano /etc/apache2/sites-available/000-default.conf
document root : /var/www/html/cafeapp

→ sudo systemctl restart apache2

↑

Setting up Web Layer

Setting up mysql layer.

→ sudo mysql -u root -p ↵ ↵
(create database, create user, create privileges

→ create database mom_pop_db.
→ create — command in doc for user creation
→ privileges - command in doc. — check db name.

Login as that user

→ sudo mysql -u msis -P → Msois@123
→ show databases : shows db
→ use mom_pop_db : goes inside the db.
→ show tables
→ source mom_pop_db/create-db.sql ;
→ show tables (4 tables)
→ select * from product (products — elements under)

Connect the database + web page

→ cd /var/www/html/cafeapp
→ ls
→ sudo nano getAppParameters.php

```
db_user :"msis";
db_url :"localhost";
```

→ sudo systemctl restart apache2

Launch EC2 Instance : without logging in - install all packages & start project (automate previous steps)

→ New instance → name : cafestaticapp
→ Ubuntu → 22.04 LTS
→ Instance type : t2.micro
→ rsckey : keypair
→ allow SSH, HTTP, HTTPS type traffic on port 443
→ 8GB gp2
→ Advanced detail → user data (add commands to be executed on start on system) ← shell script / bash script
→ Copy command / code from document. + paste
↓ runs only 1st time system boots
→ Instance → copy IP address → BROWSER (apache 2 welcome page)
→ IP address/cafeapp → static website w/o logging in.

CFA → Challenge lab

20/08

VPC

→ Web layer and database layer on different systems.

→ VPC → virtual private cloud (Isolated network)
Create a segregated space under all data centers

~~EC2~~

→ Create a VPC

→ Cafestagevpc → name

→ IPv4 CIDR block : 10.0.0.0/16 → identify unique host within network
(network identifier, subnet mask)

32 - 16 = 16 ⇒ 2^{16} 65536 → number of allowed instances (IP addresses)

→ Number of availability zones → 2.
(Multi AZ → resource spans accross 2 AZ)

→ * Public subnet → launch instance - provides a public IP address

→ * Private subnet - resource launced is accessible only within VPC.

→ Customize subnets CIDR blocks

public subnet us-east-1a
10.0.0.0/24 - launch 256 resources. ($32-24 \Rightarrow 2^8$)

public subnet us-east-1b
10.0.2.0/24
↓
identify host within network

private
10.0.1.0/24
10.0.3.0/24

→ NAT gateway : used by resources under private subnet to access internet launched inside public subnet (provides gateway)
(Internet gateway for public subnet)

→ Filter VPC – cafestagevpc

→ Subnets

→ Search EC2 – launch instance – web server

→ AMI – ubuntu – 22.04

→ Instance type : t2.micro

→ key pair name : rockey

→ Network Settings : Edit

VPC : cafestagevpc

→ Subnet : 4 (2 Private, 2 Public)

↳ public subnet : us 1a

→ Auto assign public IP : enable (enabled by default for default VPC)

the webpage to be accessible over internet

→ Inbound Security Group

Add security group

type : HTTP

Source type : Anywhere

→ Number of instance : 1

LAUNCH AN INSTANCE

→ Launch an instance

→ db-server

→ Ubuntu – 22.04

→ Network Settings – edit

VPC – cafestagevpc

→ Subnet : private us 1a

→ Auto assign : disable

LAUNCH AN INSTANCE

→ get labuser.pem from downloads
AWS details → download PEM

→ sudo chmod 600 labuser.pem
→ sudo ssh -i labuser.pem ubuntu@ public IP address (EC2, webserver)
→ yes
→ sudo apt-get update (inside webserver, install dependencies)
→ sudo apt-get install apache2 libapache2-mod-php php php-mysql mysql-client
→ sudo systemctl status apache2
→ exit

How to securely copy a file to remote server.

→ sudo scp -i labsuser.pem labsuser.pem ubuntu@ 44.223.20.233:/home/ubuntu/. (public ipv4 IP address)
(In downloads path)

copying labsuser on same name to that IP address

→ sudo ssh -i labuser.pem ubuntu@ public IP address
→ ls
→ sudo chmod 600 labuser.pem → exit (at path of webserver public IP)
→ sudo ssh -i labuser.pem ubuntu@ 10.0.1.100 (private IP address) (EC2, dbserver)
→ ip a
→ sudo apt-get update

mysql can work only on local host.

→ ip address changes everytime you login — public IP address (floating IP address)

→ sudo ssh -i labsuser.pem ubuntu@ public IP → webserver
→ sudo git clone (document command)
→ cd cafe-dynamic_website/
→ sudo mkdir /var/www/html/cafeapp (document root folder)
→ sudo cp -rf mompopcafe/* /var/www/html/cafeapp/. (copying src code to cafeapp root.)

→ sudo nano /etc/apache2/sites-available/000-default.conf

→ DocumentRoot /var/www/html/cafeapp

→ sudo systemctl restart apache2

——— webpage set ———

Do ssh to db server from web server : only possible way to access db

dbserver

→ ls

→ ssh -i labsuser.pem ubuntu@ private IP

→ sudo apt-get update

→ sudo apt-get install mysql-server mysql-client

→ sudo git clone (document command) to source sql file

→ sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf

↓ any change ↘ service to be changed

bind-address = 127.0.0.1 (local host database)

= 0.0.0.0 (or private IP of dbserver)

db listens on all IP address on the server.
(listens even if IP address changes)

→ sudo systemctl restart mysql.service

→ sudo mysql -u root -p ↵

→ create database moncafedb ;

Create a user msis who can connect via localhost + any remote IP address

→ Create USER 'msis'@'%' IDENTIFIED .... (document command)

or ip address

→ ALTER USER (doc command)

→ GRANT USER (doc command)

root user no pwd

exit from db server

→ exit (back to web server)

→ sudo mysql -u mois -h (ip address of server) -p
(db server)
host

(security group blocks the traffic - only ssh is given)

→ Webserver : security group : security group (ID)

→ dbserver : security group : Inbound rule : add a rule

→ Type : mysql , source type : Custom , Source : copied security group ID

Allows all mysql traffic from security group.

→ show database → use mom cafedb

→ source cafe-Dynamic_Website/mompopdb/create-db.sql ;

→ exit

→ cd /var/www/html/cafeapp

→ ls

→ sudo nano getAppParameters.php

db_url : private IP address (db server)

db_name : mom cafedb

db_user : mois

→ sudo systemctl restart apache2

gives error since it does have access yet (use database

Alternate on public IP go to Cafe_Dynamic_Website /mompop

Enter into sql on db

Source create-db.sql ;

→ Open website using publicIP of web server.

22/08

→ Route 53 service
→ DNS ——— type of DNS by AWS.
→ Systems manager

→ Availability zones : DNS.

cafeapp was not available to clients, owner was not aware of it. He missed few clients due to downtime.

Requirement : Inform me ASAP.
Create another route to divert traffic if one is down.
Continuosly monitor the health of service.

* VPC in all AZs & subnets in all VPC.
* Set one as primary & diff one as secondary

Route 53 : lets you use domain name instead of remembering IP address.

Failover routing policy : Route 53 uses this to re-route traffic if primary fails.

→ EC2 instance
→ VPC
→ Route 53
→ Systems Manager

→ VPC → filter mompopcafe → 2 subnets → check availability zone (public)

→ EC2 instances → 2 instances → 2 AZs.
(1 primary instance)

→ IP / mompopcafe → primary website
↳ public IP of 1 EC2 instance. / public IP of 2nd EC2 instance.

Doesn't show you which instance is rendered (only through IP)
(getAppParameter file on console)

→ AWS System Manager → Parameter store (dbuser, url, name)
instead of putting hardcoded - edit here / maintain

→ showserverinfo → parameters → edit → value true

→ Return to website → "Shows server information"

(4) Perform health check to re-route if primary fails.

→ Route 53 (AWS DNS)
→ create a healthcheck
primary-health-website
Monitor → Endpoint ; HTTP - server
IP address : 1st EC2 instance public IP.
~~host~~ port : 80
path : / mompopcafe

→ Advance configuration
Interval : 10sec
Failure threshold : 2

Next

pub & sub mechanism

(SNS - Simple Notification Service)
→ Create alarm : yes → new sms topic → health check no
Email address : your email → confirm on mail.

→ Route 53 → hosted zones → Create a record

A record → Alias : value to be translated to.
Domain to be translated to IP address

CNAME → Canonical name.

direct from one domain to another domain.

Record name : www
Record type : A record
Value : IP - public 1st EC2 instance
TTL second : 15 sec
Routing policy : failover
Record type : primary
health check : created option
Record ID : primary_one

same for 2nd EC2 instance

Change value & Record type : secondary

Use case : More time spent on administration ops.

→ Creating instances → extra ops
→ Maintaince → database manage

explain ?

⇒ Use managed database over unmanaged database

⇒ Until now we use Iaas. now Paas

⇒ RDS relational database service (managed database)

→ VPC → Create VPC → VPC and more

→ Name : Labvpc → CIDR : 10.0.0/16

→ No of AZ : 2

→ No of public subnet : 2
private : 2

→ 10.0.0.0/24, 10.0.2.0/24 } public
10.0.1.0/24, 10.0.3.0/24 } private

→ NAT → 1 AZ.

→ VPC endpoint : None

CREATE VPC

(*) Route table : decides if subnet is private/public
(route -n) : which route is taken to access internet

(*) Every subnet has only 1 Route table (compulsory)
But all subnets can be associated to 1 Route tables

[check route table under VPC]

private subnet uses nat to access internet.

→ EC2 → Launch a instance → name : web

→ ubuntu 22.04 → vokey → VPC (labvpc)

→ subnet (public SN 1)

→ Auto assign - enable (generates public IP)

→ Security : SSH, HTTP (Anywhere) My IP allows only your system to contact vi

LAUNCH AN INSTANCE

(*) Remove pem files from downloads

→ sudo rm -rf labsuser.pem

Download key : AWS details download PEM

→ sudo chmod 600 labsuser.pem

→ sudo ssh -i labsuser.pem ubuntu@ IP address — public IP

→ sudo apt-get update -y

→ sudo apt-get install apache2 mysql-client -y (install other dependencies)

→ RDS → Create a database → Standard create

→ Engine type: mysql → Engine version: latest

→ Templates : Free tier (production costs high as database is created on all AZ)

→ Settings : cafedB

→ Master username : admin → self managed

→ Masterpassword : Msois123

→ DB instance class : db t3.micro

→ Storage : General purpose : 20GB

→ Connectivity : do not connect → VPC : labvpc

→ Create new DB : public access: No

→ VPC security group → new → mydbsg → no rule

→ Tags → Name : cafedb

→ Database authentication : pwd

→ Additional Configuration [check for diff b/w managed and unmanaged DB]

DONE

→ database → cafedb → security → mydbsg → inbound rule

Source : anywhere
type : mysql

→ Database → Endpoint
(use: mysql -u admin -p -h endpoint)

→ Inside web-server (public) — remote host

→ sudo mysql -u admin -h (put domain name) -p — copy from endpoint

→ Create database : CREATE (command)

→ use database

→ grant permission - admin

→ create table

→ show table

→ exit

→ sudo mkdir /var/www/html/cafeapp

→ ls

→ sudo cp -rf mytempcafe/* /var/www/html/cafeapp

→ sudo nano /etc/apache2/sites-available/000-default → document root.

→ sudo nano getAppParameters

→ name: cafedb

→ url : db url copied

→ user: admin