

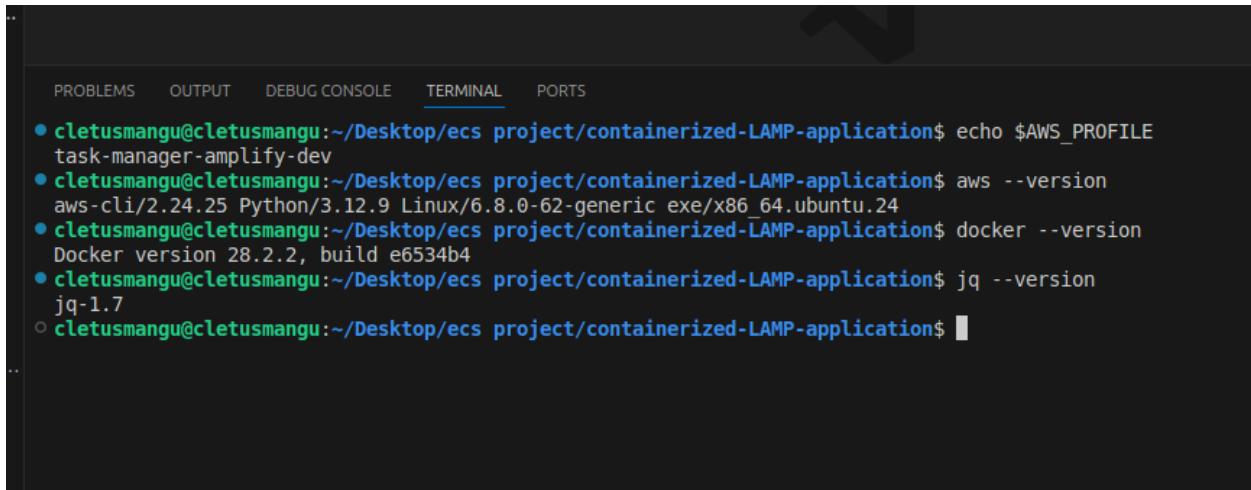
AWS CLI Deployment - LAMP Stack on ECS Fargate

These are the steps and commands used in deploying the entire containerized LAMP application using only AWS CLI commands.

Prerequisites

Required Tools

- AWS CLI v2 installed and configured
- Docker installed and running
- jq (JSON processor) for parsing CLI responses



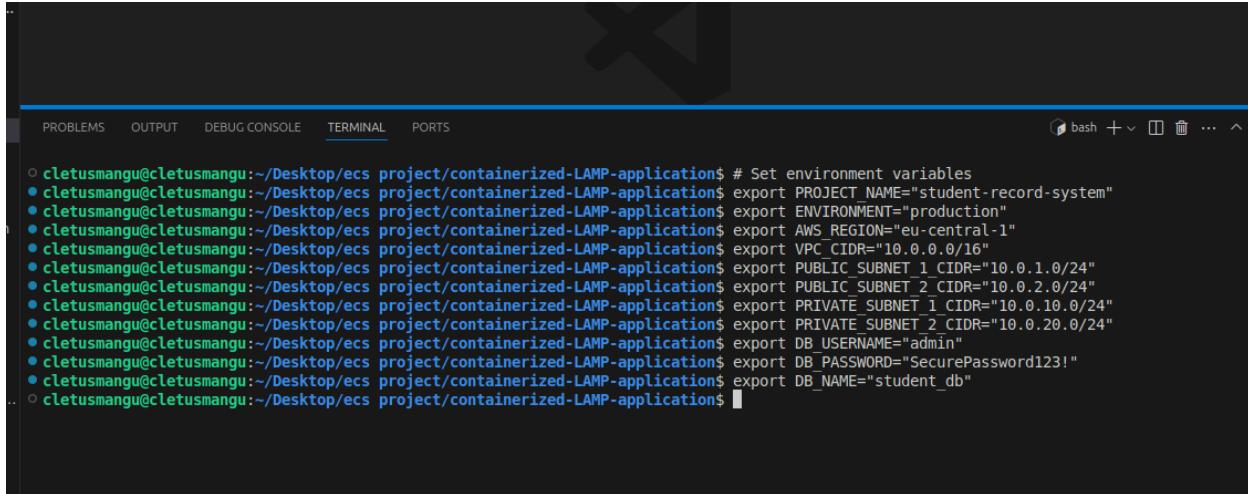
The screenshot shows a terminal window with the following command history:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo $AWS_PROFILE
task-manager-amplify-dev
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws --version
aws-cli/2.24.25 Python/3.12.9 Linux/6.8.0-62-generic exe/x86_64.ubuntu.24
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ docker --version
Docker version 28.2.2, build e6534b4
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ jq --version
jq-1.7
○ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ █
```

Environment Setup

```
# Set environment variables (customize these)
export PROJECT_NAME="student-record-system"
export ENVIRONMENT="production"
export AWS_REGION="eu-central-1"
export VPC_CIDR="10.0.0.0/16"
export PUBLIC_SUBNET_1_CIDR="10.0.1.0/24"
export PUBLIC_SUBNET_2_CIDR="10.0.2.0/24"
export PRIVATE_SUBNET_1_CIDR="10.0.10.0/24"
export PRIVATE_SUBNET_2_CIDR="10.0.20.0/24"
export DB_USERNAME="admin"
export DB_PASSWORD="SecurePassword123!" #password will change in production
```

```
export DB_NAME="student_db"
```

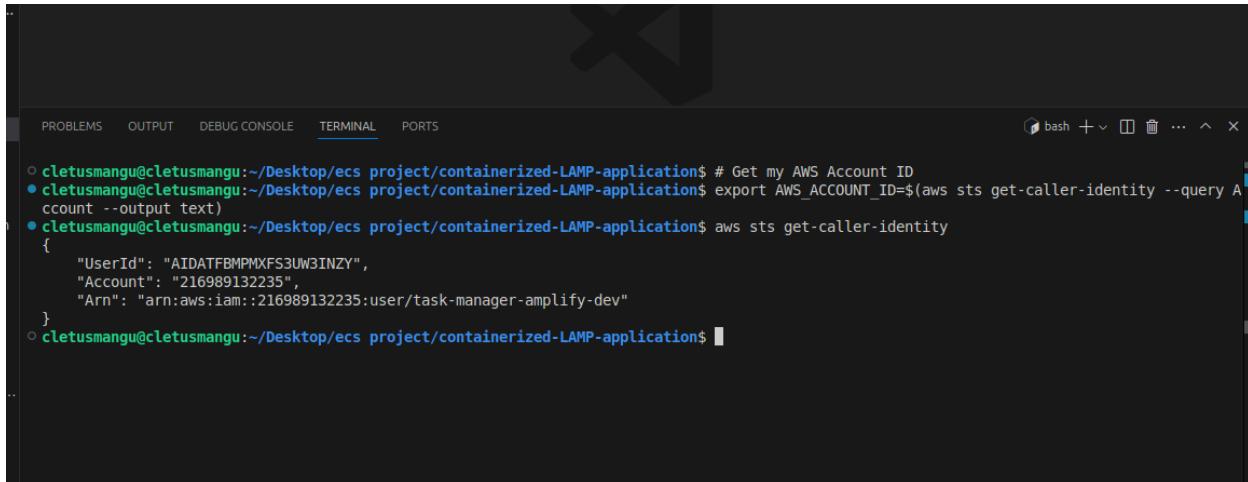


A screenshot of a terminal window titled "bash". The window shows a series of environment variable assignments:

```
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Set environment variables
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PROJECT_NAME="student-record-system"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export ENVIRONMENT="production"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export AWS_REGION="eu-central-1"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export VPC_CIDR="10.0.0.0/16"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PUBLIC_SUBNET_1_CIDR="10.0.1.0/24"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PUBLIC_SUBNET_2_CIDR="10.0.2.0/24"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PRIVATE_SUBNET_1_CIDR="10.0.10.0/24"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PRIVATE_SUBNET_2_CIDR="10.0.20.0/24"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export DB_USERNAME="admin"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export DB_PASSWORD="SecurePassword123!"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export DB_NAME="student_db"
```

```
# Get your AWS Account ID
```

```
export AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
```



A screenshot of a terminal window titled "bash". The window shows the execution of the command to get the AWS Account ID:

```
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Get my AWS Account ID
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws sts get-caller-identity
{
  "UserId": "AIDATFBMMPMXFS3UW3INZY",
  "Account": "216989132235",
  "Arn": "arn:aws:iam::216989132235:user/task-manager-amplify-dev"
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

```
# Get availability zones
```

```
export AZ1=$(aws ec2 describe-availability-zones --query 'AvailabilityZones[0].ZoneName' --output text)
```

```
export AZ2=$(aws ec2 describe-availability-zones --query 'AvailabilityZones[1].ZoneName' --output text)
```

```
echo "Account ID: $AWS_ACCOUNT_ID"
```

```
echo "Region: $AWS_REGION"
```

```
echo "AZ1: $AZ1"
```

```
echo "AZ2: $AZ2"
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash + ⊞ ⊞ ... ^ X

• cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Get availability zones
• cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export AZ1=$(aws ec2 describe-availability-zones --query 'AvailabilityZones[0].ZoneName' --output text)
• cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export AZ2=$(aws ec2 describe-availability-zones --query 'AvailabilityZones[1].ZoneName' --output text)
• cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "Account ID: $AWS_ACCOUNT_ID"
echo "Region: $AWS_REGION"
echo "AZ1: $AZ1"
echo "AZ2: $AZ2"
Account ID: 216989132235
Region: eu-central-1
AZ1: eu-central-1a
AZ2: eu-central-1b
○ cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

Phase 1: Create VPC and Networking Infrastructure

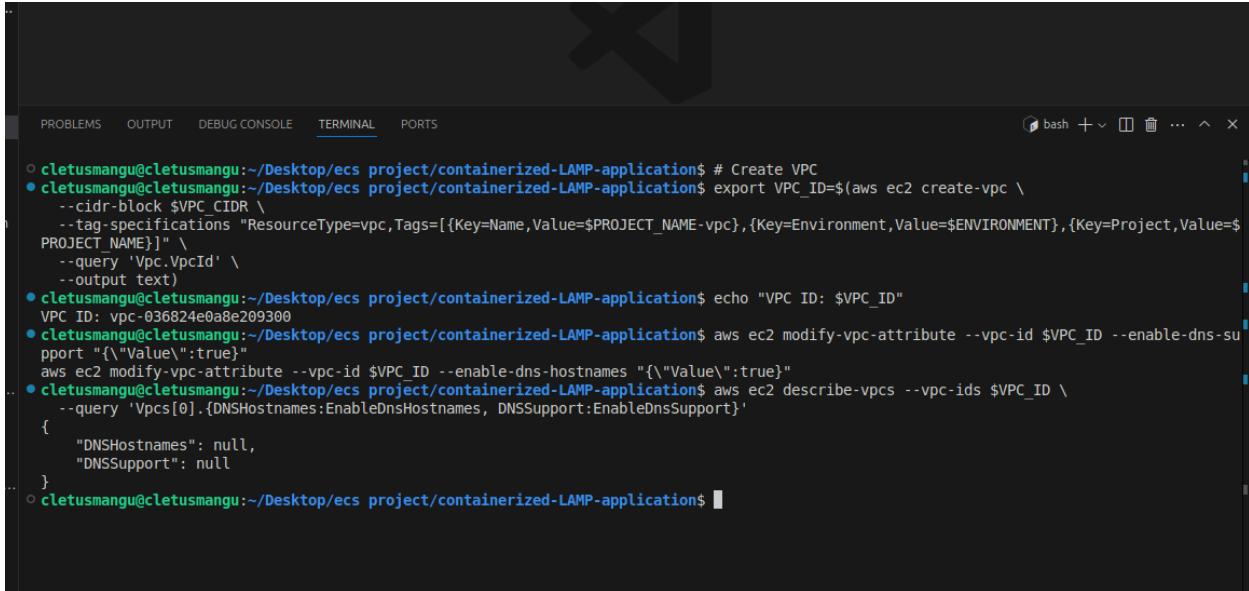
Step 1.1: Create VPC

```
# Create VPC

export VPC_ID=$(aws ec2 create-vpc \
--cidr-block $VPC_CIDR \
--enable-dns-hostnames \
--enable-dns-support \
--tag-specifications "ResourceType=vpc,Tags=[{Key=Name,Value=$PROJECT_NAME-vpc}, {Key=Environment,Value=$ENVIRONMENT}, {Key=Project,Value=$PROJECT_NAME}]" \
--query 'Vpc.VpcId' \
--output text) echo "VPC ID: $VPC_ID"

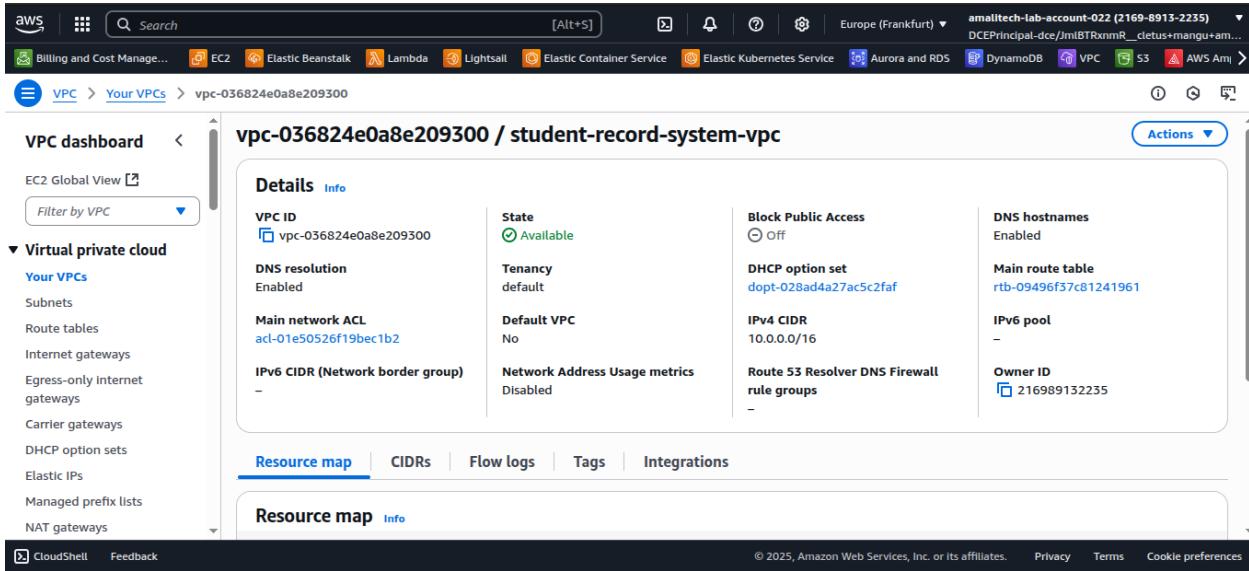
# Wait for VPC to be available

aws ec2 wait vpc-available --vpc-ids $VPC_ID
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash + ⌂ ⌂ ... ^ ×

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Create VPC
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export VPC_ID=$(aws ec2 create-vpc \
  --cidr-block $VPC_CIDR \
  --tag-specifications "ResourceType=vpc,Tags=[{Key=Name,Value=$PROJECT_NAME-vpc},{Key=Environment,Value=$ENVIRONMENT},{Key=Project,Value=$PROJECT_NAME}]" \
  --query 'Vpc.VpcId' \
  --output text)
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "VPC ID: $VPC_ID"
VPC ID: vpc-036824e0a8e209300
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ec2 modify-vpc-attribute --vpc-id $VPC_ID --enable-dns-support "{\"Value\":true}"
aws ec2 modify-vpc-attribute --vpc-id $VPC_ID --enable-dns-hostnames "{\"Value\":true}"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ec2 describe-vpcs --vpc-ids $VPC_ID \
  --query 'Vpcs[0].{DNSHostnames:EnableDnsHostnames, DNSSupport:EnableDnsSupport}'
{
  "DNSHostnames": null,
  "DNSSupport": null
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```



The screenshot shows the AWS VPC dashboard for the VPC `vpc-036824e0a8e209300 / student-record-system-vpc`. The **Details** tab is selected, displaying the following configuration:

VPC ID	State	Block Public Access	DNS hostnames
vpc-036824e0a8e209300	Available	Off	Enabled
DNS resolution	Tenancy	DHCP option set	Main route table
Enabled	default	dopt-028ad4a27ac5c2faf	rtb-0949ef37c81241961
Main network ACL	Default VPC	IPv4 CIDR	IPv6 pool
acl-01e50526f19bec1b2	No	10.0.0.0/16	-
IPv6 CIDR (Network border group)	Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID
-	Disabled	-	216989132235

Below the main table, there are tabs for **Resource map**, **CIDRs**, **Flow logs**, **Tags**, and **Integrations**. The **Resource map** tab is also visible.

Step 1.2: Create Internet Gateway

```
# Create Internet Gateway

export IGW_ID=$(aws ec2 create-internet-gateway \
  --tag-specifications "ResourceType=internet-
  gateway,Tags=[{Key=Name,Value=$PROJECT_NAME-
  igw},{Key=Environment,Value=$ENVIRONMENT}]" \
```

```
--query 'InternetGateway.InternetGatewayId' \
--output text) echo "Internet Gateway ID: $IGW_ID"

# Attach Internet Gateway to VPC

aws ec2 attach-internet-gateway \
--internet-gateway-id $IGW_ID \
--vpc-id $VPC_ID
```

The terminal window shows the following session:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash + ... ^ x

○ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Create Internet Gateway
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export IGW_ID=$(aws ec2 create-internet-gateway \
    --tag-specifications "ResourceType=internet-gateway,Tags=[{Key=Name,Value=$PROJECT_NAME-igw},{Key=Environment,Value=$ENVIRONMENT}]" \
    --query 'InternetGateway.InternetGatewayId' \
    --output text)
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "Internet Gateway ID: $IGW_ID"
Internet Gateway ID: igw-0dcf1fe6affd5b69b
○ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Attach Internet Gateway to VPC
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ec2 attach-internet-gateway \
    --internet-gateway-id $IGW_ID \
    --vpc-id $VPC_ID
○ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

The screenshot shows the AWS VPC service page with the following details for the Internet Gateway:

- Internet gateway ID:** igw-0dcf1fe6affd5b69b
- State:** Attached
- VPC ID:** vpc-036824e0a8e209300 | student-record-system-vpc
- Owner:** 216989152235

The **Tags** section lists two tags:

Key	Value
Environment	production
Name	student-record-system-igw

Step 1.3: Create Subnets

```
# Create Public Subnet 1

export PUBLIC_SUBNET_1_ID=$(aws ec2 create-subnet \
```

```
--vpc-id $VPC_ID \
--cidr-block $PUBLIC_SUBNET_1_CIDR \
--availability-zone $AZ1 \
--map-public-ip-on-launch \
--tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-public-subnet-1}, {Key=Type,Value=Public},{Key=Environment,Value=$ENVIRONMENT}]"\ \
--query 'Subnet.SubnetId' \
--output text)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash + v ⌂ ... ^ x

○ cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Create Public Subnet 1
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PUBLIC_SUBNET_1_ID=$(aws ec2 create-subnet \
  --vpc-id $VPC_ID \
  --cidr-block $PUBLIC_SUBNET_1_CIDR \
  --availability-zone $AZ1 \
  --tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-public-subnet-1}, {Key=Type,Value=Public},{Key=Environment,Value=$ENVIRONMENT}]"\ \
  --query 'Subnet.SubnetId' \
  --output text)
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ec2 modify-subnet-attribute \
  --subnet-id $PUBLIC_SUBNET_1_ID \
  --map-public-ip-on-launch
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ec2 describe-subnets \
  --subnet-ids $PUBLIC_SUBNET_1_ID \
  --query 'Subnets[0].MapPublicIpOnLaunch'
true
○ cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

The screenshot shows the AWS VPC Subnets page with the following details for the subnet:

- Subnet ID:** subnet-0cffb475ccc65f5a1
- IPv4 CIDR:** 10.0.1.0/24
- Availability Zone:** eu-central-1a
- Route table:** rtb-09496f37c81241961
- Auto-assign IPv6 address:** No
- IPv4 CIDR reservations:** -
- Resource name DNS A record:** Disabled
- Subnet ARN:** arn:aws:ec2:eu-central-1:216989132235:subnet/subnet-0cffb475ccc65f5a1
- State:** Available
- IPv6 CIDR:** -
- Available IPv4 addresses:** 251
- Availability Zone ID:** euc1-a2
- Network ACL:** acl-01e50526f19bec1b2
- Auto-assign customer-owned IPv4 address:** No
- IPv6 CIDR reservations:** -
- Block Public Access:** Off
- IPv6 CIDR association ID:** -
- VPC:** vpc-036824e0a8e209300 | student-record-system-vpc
- Auto-assign public IPv4 address:** Yes
- Outpost ID:** -
- Hostname type:** IP name
- Owner:** cletusmangu

Create Public Subnet 2

```

export PUBLIC_SUBNET_2_ID=$(aws ec2 create-subnet \
--vpc-id $VPC_ID \
--cidr-block $PUBLIC_SUBNET_2_CIDR \
--availability-zone $AZ2 \
--map-public-ip-on-launch \
--tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-public-subnet-2}, {Key=Type,Value=Public}, {Key=Environment,Value=$ENVIRONMENT}]" \
--query 'Subnet.SubnetId' \
--output text)

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash + ⊞ ... ^ ×

cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-applications$ # Create Public Subnet
• cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-application$ export PUBLIC_SUBNET_2_ID=$(aws ec2 create-subnet \
--vpc-id $VPC_ID \
--cidr-block $PUBLIC_SUBNET_2_CIDR \
--availability-zone $AZ2 \
--tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-public-subnet-2}, {Key=Type,Value=Public}, {Key=Environment,Value=$ENVIRONMENT}]" \
--query 'Subnet.SubnetId' \
--output text)
• cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ec2 modify-subnet-attribute \
--subnet-id $PUBLIC_SUBNET_2_ID \
--map-public-ip-on-launch
• cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ec2 describe-subnets \
--subnet-ids $PUBLIC_SUBNET_2_ID \
--query 'Subnets[0].MapPublicIpOnLaunch'
true
cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-application$ 

```

The screenshot shows the AWS VPC Subnets page with the following details for the subnet:

- Details** section:
 - Subnet ID:** subnet-04f8c597c03872f64
 - Subnet ARN:** arn:aws:ec2:eu-central-1:21698913235:subnet/subnet-04f8c597c03872f64
 - State:** Available
 - IPv6 CIDR:** -
 - Available IPv4 addresses:** 251
 - Availability Zone:** eu-central-1b
 - Route table:** rtb-09496f37c81241961
 - Auto-assign IPv6 address:** No
 - IPv4 CIDR reservations:** -
 - Auto-assign customer-owned IPv4 address:** No
 - IPv6 CIDR reservations:** -
- Actions** button is visible at the top right.
- Block Public Access:** Off
- IPv6 CIDR association ID:** -
- VPC:** vpc-036824e0a8e209500 | student-record-system-vpc
- Auto-assign public IPv4 address:** Yes
- Outpost ID:** -
- Hostname type:** IP name
- Owner:** Owner

```
# Create Private Subnet 1

export PRIVATE_SUBNET_1_ID=$(aws ec2 create-subnet \
--vpc-id $VPC_ID \
--cidr-block $PRIVATE_SUBNET_1_CIDR \
--availability-zone $AZ1 \
--tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-private-subnet-1}, {Key=Type,Value=Private}, {Key=Environment,Value=$ENVIRONMENT}]" \
--query 'Subnet.SubnetId' \
--output text)
```

The screenshot shows a terminal window with the following session:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

```
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Create Private Subnet 1
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PRIVATE_SUBNET_1_ID=$(aws ec2 create-subnet \
  --vpc-id $VPC_ID \
  --cidr-block $PRIVATE_SUBNET_1_CIDR \
  --availability-zone $AZ1 \
  --tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-private-subnet-1}, {Key=Type,Value=Private}, {Key=Environment,Value=$ENVIRONMENT}]" \
  --query 'Subnet.SubnetId' \
  --output text)
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "Private Subnet 1 ID: $PRIVATE_SUBNET_1_ID"
Private Subnet 1 ID: $PRIVATE_SUBNET_1_ID
aws ec2 describe-subnets \
  --subnet-ids $PRIVATE_SUBNET_1_ID \
  --query 'Subnets[0].[ID:SubnetId, AZ:AvailabilityZone, CIDR:CidrBlock, PublicIP:MapPublicIpOnLaunch]'

{
  "ID": "subnet-0b269f9387e9cd051",
  "AZ": "eu-central-1a",
  "CIDR": "10.0.10.0/24",
  "PublicIP": false
}
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

The screenshot shows the AWS VPC Subnets page. The top navigation bar includes links for Billing and Cost Management, EC2, Elastic Beanstalk, Lambda, Lightsail, Elastic Container Service, Elastic Kubernetes Service, Aurora and RDS, DynamoDB, VPC, S3, and AWS Amplify. The main content area displays the details of a specific subnet:

Details	
Subnet ID	<code>subnet-0b269f9387e9cd051</code>
Subnet ARN	<code>arn:aws:ec2:eu-central-1:216989152235:subnet/subnet-0b269f9387e9cd051</code>
State	Available
IPv4 CIDR	<code>10.0.10.0/24</code>
IPv6 CIDR	—
Availability Zone	<code>eu-central-1a</code>
Available IPv4 addresses	251
Availability Zone ID	<code>euc1-az2</code>
Network border group	<code>eu-central-1</code>
Route table	<code>rtb-09496f37c1241961</code>
Default subnet	No
Auto-assign IPv6 address	No
Customer-owned IPv4 pool	—
IPv4 CIDR reservations	—
IPv6 CIDR reservations	—
IPv6-only	No
DNS64	Disabled
Resource name DNS A record	Disabled
Outpost ID	—
Hostname type	IP name
Owner	amalitech-lab-account-022 (2169-8913-2235)
Block Public Access	Off
IPv6 CIDR association ID	—
VPC	<code>vpc-036824e0a8e209300</code> student-record-system-vpc
Auto-assign public IPv4 address	No

At the bottom of the page, there are links for CloudShell, Feedback, and footer information: © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

Create Private Subnet 2

```
export PRIVATE_SUBNET_2_ID=$(aws ec2 create-subnet \
--vpc-id $VPC_ID \
--cidr-block $PRIVATE_SUBNET_2_CIDR \
--availability-zone $AZ2 \
--tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-private-subnet-2}, {Key=Type,Value=Private}, {Key=Environment,Value=$ENVIRONMENT}]"\ \
--query 'Subnet.SubnetId' \
--output text)
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Create Private Subnet 2
export PRIVATE_SUBNET_2_ID=$(aws ec2 create-subnet \
    --vpc-id $VPC_ID \
    --cidr-block $PRIVATE_SUBNET_2_CIDR \
    --availability-zone $AZ2 \
    --tag-specifications "ResourceType=subnet,Tags=[{Key=Name,Value=$PROJECT_NAME-private-subnet-2},{Key=Type,Value=Private},{Key=Environment,Value=$ENVIRONMENT}]" \
    --query 'Subnet.SubnetId' \
    --output text)
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "Private Subnet 2 ID: $PRIVATE_SUBNET_2_ID"
aws ec2 describe-subnets \
    --subnet-ids $PRIVATE_SUBNET_2_ID \
    --query 'Subnets[0].[ID:SubnetId, AZ:AvailabilityZone, CIDR:CidrBlock, PublicIP:MapPublicIpOnLaunch]'
Private Subnet 2 ID: subnet-015c9977de182e46d
{
    "ID": "subnet-015c9977de182e46d",
    "AZ": "eu-central-1b",
    "CIDR": "10.0.20.0/24",
    "PublicIP": false
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ 

```

The screenshot shows the AWS VPC Subnets page for the subnet `subnet-015c9977de182e46d`. The subnet is part of the `student-record-system-private-subnet-2` and is located in the `eu-central-1b` availability zone. It has an IPv4 CIDR of `10.0.20.0/24` and is associated with the route table `rtb-09496f37c81241961`. The subnet is currently `Available` and has 251 available IPv4 addresses. It is not connected to any network border groups or customer-owned IPv4 pools. The `Block Public Access` setting is off. The `VPC` identifier is `vpc-036824e0a8e209300`. The `Owner` is listed as `amalitech-lab-account-022 (2169-8913-2235)`.

```

echo "Public Subnet 1: $PUBLIC_SUBNET_1_ID"
echo "Public Subnet 2: $PUBLIC_SUBNET_2_ID"
echo "Private Subnet 1: $PRIVATE_SUBNET_1_ID"
echo "Private Subnet 2: $PRIVATE_SUBNET_2_ID"

```

Step 1.4: Create NAT Gateways

Allocate Elastic IPs for NAT Gateways

```
export EIP_1_ALLOCATION_ID=$(aws ec2 allocate-address \
--domain vpc \
--tag-specifications "ResourceType=elastic-
ip,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-eip-1},
{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'AllocationId' \
--output text) export EIP_2_ALLOCATION_ID=$(aws ec2 allocate-address \
--domain vpc \
--tag-specifications "ResourceType=elastic-
ip,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-eip-
2},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'AllocationId' \
--output text)
```

The screenshot shows a terminal window with several command-line sessions. The sessions are as follows:

- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Allocate Elastic IPs for NAT Gateways`
- `export EIP_1_ALLOCATION_ID=$(aws ec2 allocate-address \
--domain vpc \
--tag-specifications "ResourceType=elastic-ip,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-eip-1},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'AllocationId' \
--output text)`
- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export EIP_2_ALLOCATION_ID=$(aws ec2 allocate-address \
--domain vpc \
--tag-specifications "ResourceType=elastic-ip,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-eip-2},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'AllocationId' \
--output text)`
- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Create NAT Gateways`
- `export NAT_GW_1_ID=$(aws ec2 create-nat-gateway \
--subnet-id $PUBLIC_SUBNET_1_ID \
--allocation-id $EIP_1_ALLOCATION_ID \
--tag-specifications "ResourceType=nat-gateway,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-gw-1},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'NatGateway.NatGatewayId' \
--output text)`
- An error occurred (InvalidParameter) when calling the CreateNatGateway operation: 'nat-gateway' is not a valid taggable resource type for this operation.
- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export NAT_GW_2_ID=$(aws ec2 create-nat-gateway \
--subnet-id $PUBLIC_SUBNET_2_ID \
--allocation-id $EIP_2_ALLOCATION_ID \
--tag-specifications "ResourceType=nat-gateway,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-gw-2},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'NatGateway.NatGatewayId' \
--output text)`
- An error occurred (InvalidParameter) when calling the CreateNatGateway operation: 'nat-gateway' is not a valid taggable resource type for this operation.
- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "NAT Gateway 1: $NAT_GW_1_ID"
NAT Gateway 1:
NAT Gateway 2:
NAT Gateway 2:`

Create NAT Gateways

```
export NAT_GW_1_ID=$(aws ec2 create-nat-gateway \
--subnet-id $PUBLIC_SUBNET_1_ID \
--allocation-id $EIP_1_ALLOCATION_ID \
--tag-specifications "ResourceType=nat-
gateway,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-gw-
1},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'NatGateway.NatGatewayId' \
--output text) export NAT_GW_2_ID=$(aws ec2 create-nat-gateway \
--subnet-id $PUBLIC_SUBNET_2_ID \
--allocation-id $EIP_2_ALLOCATION_ID \
--tag-specifications "ResourceType=nat-
gateway,Tags=[{Key=Name,Value=$PROJECT_NAME-nat-gw-
2},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'NatGateway.NatGatewayId' \
--output text) echo "NAT Gateway 1: $NAT_GW_1_ID" echo "NAT Gateway 2:
$NAT_GW_2_ID"
```

Wait for NAT Gateways to be available

```
aws ec2 wait nat-gateway-available --nat-gateway-ids $NAT_GW_1_ID
$NAT_GW_2_ID
```

nat-00398d6a0323d449c

Details		Actions	
NAT gateway ID nat-00398d6a0323d449c	Connectivity type Public	State Available	State message —
NAT gateway ARN arn:aws:ec2:eu-central-1:216989132235:natgateway/nat-00398d6a0323d449c	Primary public IPv4 address 3.70.56.96	Primary private IPv4 address 10.0.2.191	Primary network interface ID eni-0917cc064d9f2b355
VPC vpc-036824e0a8e209300 / student-record-system-vpc	Subnet subnet-04f8c597c03872f64 / student-record-system-public-subnet-2	Created Saturday, July 5, 2025 at 09:19:20 GMT	Deleted —

Secondary IPv4 addresses | Monitoring | Tags

nat-02299267ba45fce5d

Details		Actions	
NAT gateway ID nat-02299267ba45fce5d	Connectivity type Public	State Available	State message —
NAT gateway ARN arn:aws:ec2:eu-central-1:216989132235:natgateway/nat-02299267ba45fce5d	Primary public IPv4 address 18.197.98.190	Primary private IPv4 address 10.0.1.136	Primary network interface ID eni-077599e98c0ba2458
VPC vpc-036824e0a8e209300 / student-record-system-vpc	Subnet subnet-0cffb475ccc65f5a1 / student-record-system-public-subnet-1	Created Saturday, July 5, 2025 at 09:19:19 GMT	Deleted —

Secondary IPv4 addresses | Monitoring | Tags

Step 1.5: Create Route Tables

```
# Create Public Route Table
export PUBLIC_RT_ID=$(aws ec2 create-route-table \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=route-
table,Tags=[{Key=Name,Value=$PROJECT_NAME-public-
rt},{Key=Environment,Value=$ENVIRONMENT}]" \
```

```
--query 'RouteTable.RouteTableId' \
--output text)
```

The terminal window shows the following AWS CLI commands:

```
cetusmangu@cetusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PUBLIC_RT_ID=$(aws ec2 create-route-table \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=route-table,Tags=[{Key=Name,Value=$PROJECT_NAME-public-rt},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'RouteTable.RouteTableId' \
--output text)
cetusmangu@cetusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "Public Route Table ID: $PUBLIC_RT_ID"
```

The AWS VPC console screenshot shows the details of the route table 'rtb-03c202a4bdce23477' in the 'student-record-system-public-rt' VPC. The 'Details' tab shows the route table ID, VPC (vpc-036824e0a8e209300), and owner ID (216989132235). The 'Routes' tab shows one route entry:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

```
# Create Private Route Tables
export PRIVATE_RT_1_ID=$(aws ec2 create-route-table \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=route-
table,Tags=[{Key=Name,Value=$PROJECT_NAME-private-rt-
1},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'RouteTable.RouteTableId' \
--output text)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PRIVATE_RT_1_ID=$(aws ec2 create-route-table \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=route-table,Tags=[{Key=Name,Value=$PROJECT_NAME-private-rt-1},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'RouteTable.RouteTableId' \
--output text)
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "Private Route Table 1 ID: $PRIVATE_RT_1_ID"
aws ec2 describe-route-tables \
--route-table-ids $PRIVATE_RT_1_ID \
--query 'RouteTables[0].Tags'
Private Route Table 1 ID: rtb-0bf1dc6e65106cb0d
[
  {
    "Key": "Name",
    "Value": "student-record-system-private-rt-1"
  },
  {
    "Key": "Environment",
    "Value": "production"
  }
]
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

aws Billing and Cost Management EC2 Elastic Beanstalk Lambda Lightsail Elastic Container Service Elastic Kubernetes Service Aurora and RDS DynamoDB VPC S3 AWS Amplify

VPC > Route tables > rtb-0bf1dc6e65106cb0d

rtb-0bf1dc6e65106cb0d / student-record-system-private-rt-1

Details Info

Route table ID rtb-0bf1dc6e65106cb0d	Main No	Explicit subnet associations -	Edge associations -
VPC vpc-036824e0a8e209300 student-record-system-vpc	Owner ID 216989132235		

Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Both				Edit routes
Destination	Target	Status	Propagated	
10.0.0.0/16	local	Active	No	

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```
export PRIVATE_RT_2_ID=$(aws ec2 create-route-table \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=route-
table,Tags=[{Key=Name,Value=$PROJECT_NAME-private-rt-
2},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'RouteTable.RouteTableId' \
--output text)
```

The screenshot shows a terminal session and the AWS VPC Route Tables console.

Terminal Session:

```

$ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export PRIVATE_RT_2_ID=$(aws ec2 create-route-table \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=route-table,Tags=[{Key=Name,Value=$PROJECT_NAME-private-rt-2},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'RouteTable.RouteTableId' \
--output text)
$ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "Private Route Table 2 ID: $PRIVATE_RT_2_ID"
aws ec2 describe-route-tables \
--route-table-ids $PRIVATE_RT_2_ID \
--query 'RouteTables[0].Tags'
Private Route Table 2 ID: rtb-071ec3a2649ed8366
[
    {
        "Key": "Environment",
        "Value": "production"
    },
    {
        "Key": "Name",
        "Value": "student-record-system-private-rt-2"
    }
]
$ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ 

```

AWS VPC Route Tables Console:

The URL is [https://console.aws.amazon.com/vpc/route-tables/rtb-071ec3a2649ed8366](#)

Details Info:

Route table ID rtb-071ec3a2649ed8366	Main No	Explicit subnet associations —	Edge associations —
VPC vpc-036824e0a8e209300 student-record-system-vpc	Owner ID 216989132235		

Routes:

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

Create Routes

```
aws ec2 create-route \
--route-table-id $PUBLIC_RT_ID \
--destination-cidr-block 0.0.0.0/0 \
--gateway-id $IGW_ID
```

```
aws ec2 create-route \
--route-table-id $PRIVATE_RT_1_ID \
--destination-cidr-block 0.0.0.0/0 \
--nat-gateway-id $NAT_GW_1_ID
```

```
aws ec2 create-route \
--route-table-id $PRIVATE_RT_2_ID \
--destination-cidr-block 0.0.0.0/0 \
--nat-gateway-id $NAT_GW_2_ID
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Create Routes
aws ec2 create-route \
--route-table-id $PUBLIC_RT_ID \
--destination-cidr-block 0.0.0.0/0 \
--gateway-id $IGW_ID

aws ec2 create-route \
--route-table-id $PRIVATE_RT_1_ID \
--destination-cidr-block 0.0.0.0/0 \
--nat-gateway-id $NAT_GW_1_ID

aws ec2 create-route \
--route-table-id $PRIVATE_RT_2_ID \
--destination-cidr-block 0.0.0.0/0 \
--nat-gateway-id $NAT_GW_2_ID
{
    "Return": true
}
{
    "Return": true
}
{
    "Return": true
}
{
    "Return": true
}

cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

Associate Route Tables with Subnets

```
aws ec2 associate-route-table \
--route-table-id $PUBLIC_RT_ID \
--subnet-id $PUBLIC_SUBNET_1_ID
```

```
aws ec2 associate-route-table \
--route-table-id $PUBLIC_RT_ID \
--subnet-id $PUBLIC_SUBNET_2_ID
```

```
aws ec2 associate-route-table \
--route-table-id $PRIVATE_RT_1_ID \
--subnet-id $PRIVATE_SUBNET_1_ID
```

```
aws ec2 associate-route-table \
--route-table-id $PRIVATE_RT_2_ID \
--subnet-id $PRIVATE_SUBNET_2_ID
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Associate Route Tables with Subnets
aws ec2 associate-route-table \
--route-table-id $PUBLIC_RT_ID \
--subnet-id $PUBLIC_SUBNET_1_ID

aws ec2 associate-route-table \
--route-table-id $PUBLIC_RT_ID \
--subnet-id $PUBLIC_SUBNET_2_ID

aws ec2 associate-route-table \
--route-table-id $PRIVATE_RT_1_ID \
--subnet-id $PRIVATE_SUBNET_1_ID

aws ec2 associate-route-table \
--route-table-id $PRIVATE_RT_2_ID \
--subnet-id $PRIVATE_SUBNET_2_ID
{
    "AssociationId": "rtbassoc-0badb9e552bla44a5",
    "AssociationState": {
        "State": "associated"
    }
}
{
    "AssociationId": "rtbassoc-0a16fbf57af30ab7f",
    "AssociationState": {
        "State": "associated"
    }
}
{
    "AssociationId": "rtbassoc-02cfcaa5419d29b130",
    "AssociationState": {
        "State": "associated"
    }
}

```

Phase 2: Create Security Groups

Step 2.1: Application Load Balancer Security Group

```

export ALB_SG_ID=$(aws ec2 create-security-group \
--group-name "$PROJECT_NAME-alb-sg" \
--description "Security group for Application Load Balancer" \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=security-
group,Tags=[{Key=Name,Value=$PROJECT_NAME-alb-sg},
{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'GroupId' \
--output text)

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash + ↻ ⌂ ... ^ ×
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export ALB_SG_ID=$(aws ec2 create-security-group \
--group-name "$PROJECT_NAME-alb-sg" \
--description "Security group for Application Load Balancer" \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=security-group,Tags=[{Key=Name,Value=$PROJECT_NAME-alb-sg},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'GroupId' \
--output text)
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Add rules to ALB Security Group
aws ec2 authorize-security-group-ingress \
--group-id $ALB_SG_ID \
--protocol tcp \
--port 80 \
--cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress \
--group-id $ALB_SG_ID \
--protocol tcp \
--port 443 \
--cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0c9ab7434e89757c9",
      "GroupId": "sg-07ab79ae478a3a79e",
      "GroupOwnerId": "216989132235",
      "Istgress": false,
      "IpProtocol": "tcp",
      "FromPort": 80,
      "ToPort": 80,
      "CidrIpv4": "0.0.0.0/0",
      "SecurityGroupRuleArn": "arn:aws:ec2:eu-central-1:216989132235:security-group-rule/sgr-0c9ab7434e89757c9"
}

```

Add rules to ALB Security Group aws ec2 authorize-security-group-ingress \

--group-id \$ALB_SG_ID \

--protocol tcp \

--port 80 \

--cidr 0.0.0.0/0 aws ec2 authorize-security-group-ingress \

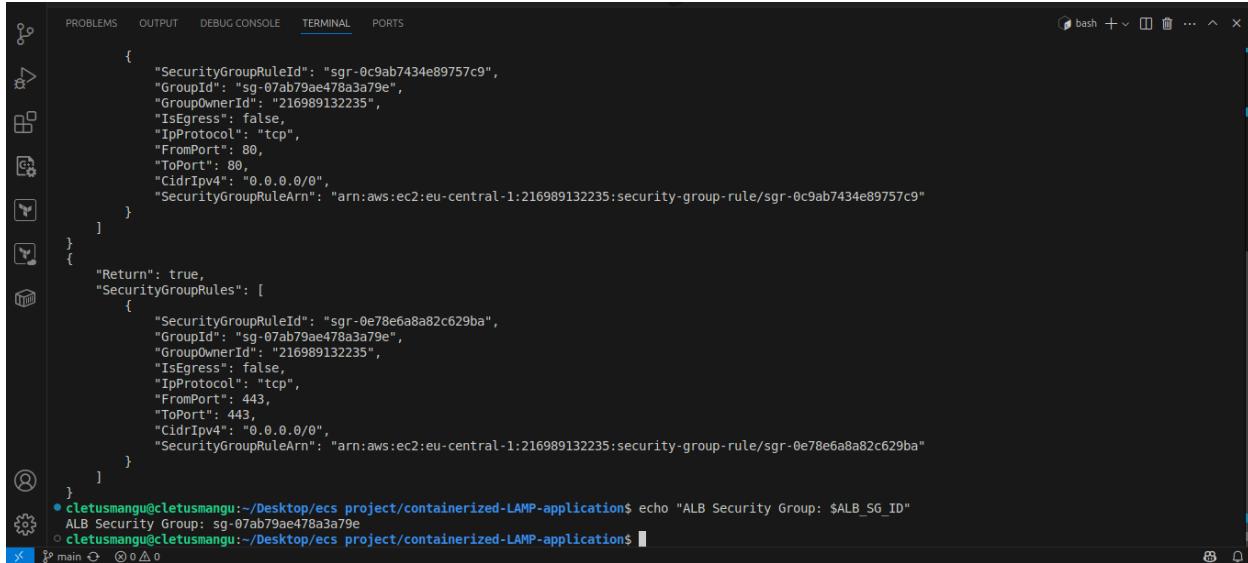
--group-id \$ALB_SG_ID \

--protocol tcp \

--port 443 \

--cidr 0.0.0.0/0

echo "ALB Security Group: \$ALB_SG_ID"



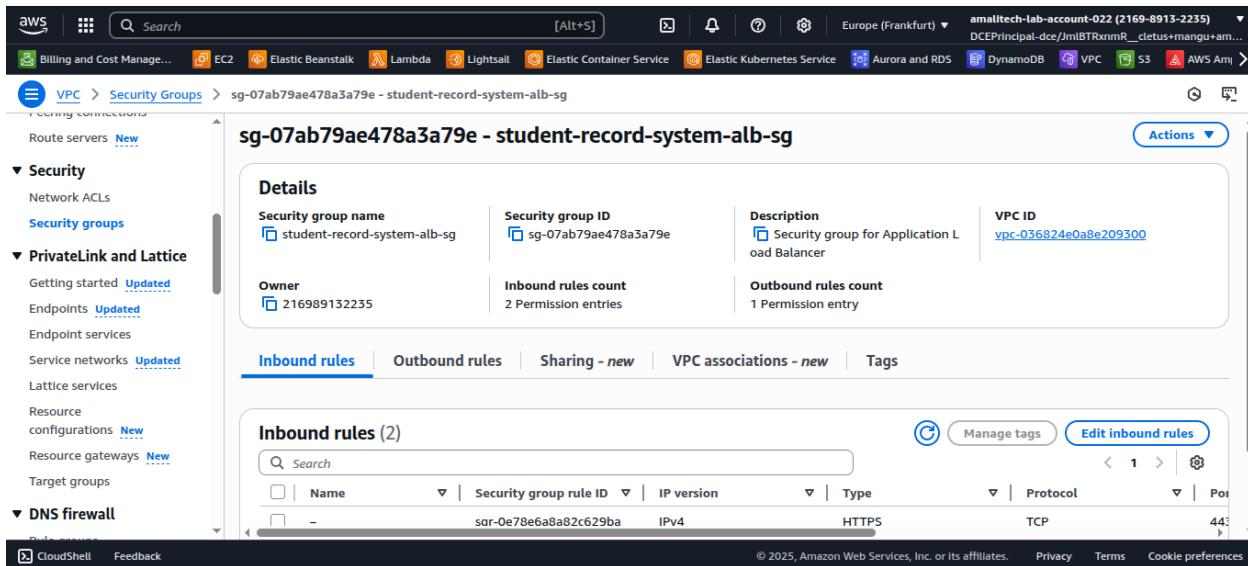
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash + × └ ... ^ ×

{
  "SecurityGroupRuleId": "sgr-0c9ab7434e89757c9",
  "GroupId": "sg-07ab79ae478a3a79e",
  "GroupOwnerId": "216989132235",
  "IsEgress": false,
  "IpProtocol": "tcp",
  "FromPort": 80,
  "ToPort": 80,
  "CidrIpv4": "0.0.0.0/0",
  "SecurityGroupRuleArn": "arn:aws:ec2:eu-central-1:216989132235:security-group-rule/sgr-0c9ab7434e89757c9"
}
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0e78e6a8a82c629ba",
      "GroupId": "sg-07ab79ae478a3a79e",
      "GroupOwnerId": "216989132235",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 443,
      "ToPort": 443,
      "CidrIpv4": "0.0.0.0/0",
      "SecurityGroupRuleArn": "arn:aws:ec2:eu-central-1:216989132235:security-group-rule/sgr-0e78e6a8a82c629ba"
    }
  ]
}

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ echo "ALB Security Group: $ALB_SG_ID"
ALB Security Group: sg-07ab79ae478a3a79e
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ 

```



The screenshot shows the AWS VPC Security Groups console. On the left, there's a navigation sidebar with sections like Security, PrivateLink and Lattice, and DNS Firewall. The main area displays the details of a security group named 'sg-07ab79ae478a3a79e - student-record-system-alb-sg'. The 'Details' tab is selected, showing information such as the security group name ('student-record-system-alb-sg'), security group ID ('sg-07ab79ae478a3a79e'), description ('Security group for Application Load Balancer'), owner ('216989132235'), and inbound/outbound rule counts. Below the details, there are tabs for Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. Under the Inbound rules tab, a table lists two rules: one for port 443 (HTTPS) and another for port 80 (TCP). At the bottom of the page, there are links for CloudShell and Feedback.

Step 2.2: ECS Tasks Security Group

```

export ECS_SG_ID=$(aws ec2 create-security-group \
--group-name "$PROJECT_NAME-ecs-sg" \
--description "Security group for ECS tasks" \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=security-
group,Tags=[{Key=Name,Value=$PROJECT_NAME-ecs-sg},
{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'GroupId' \

```

--output text)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export ECS_SG_ID=$(aws ec2 create-security-group \
--group-name "$PROJECT_NAME-ecs-sg" \
--description "Security group for ECS tasks" \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=security-group,Tags=[{Key=Name,Value=$PROJECT_NAME-ecs-sg},{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'GroupId' \
--output text)
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Add rule to allow traffic from ALB
aws ec2 authorize-security-group-ingress \
--group-id $ECS_SG_ID \
--protocol tcp \
--port 80 \
--source-group $ALB_SG_ID
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-01d3ad50ba5deb8ba",
      "GroupId": "sg-0b0d10be63d13dd9d",
      "GroupOwnerId": "216989132235",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 80,
      "ToPort": 80,
      "ReferencedGroupInfo": {
        "GroupId": "sg-07ab79ae478a3a79e",
        "UserId": "216989132235"
      },
      "SecurityGroupRuleArn": "arn:aws:ec2:eu-central-1:216989132235:security-group-rule/sgr-01d3ad50ba5deb8ba"
    }
  ]
}
$ main ⇧ 0 △ 0
```

Add rule to allow traffic from ALB

```
aws ec2 authorize-security-group-ingress \
```

```
--group-id $ECS_SG_ID \
```

```
--protocol tcp \
```

```
--port 80 \
```

```
--source-group $ALB_SG_ID
```

```
echo "ECS Security Group: $ECS_SG_ID"
```

The screenshot shows the AWS VPC Security Groups console. On the left, there's a navigation sidebar with links like Billing and Cost Management, EC2, Elastic Beanstalk, Lambda, Lightsail, Elastic Container Service, Elastic Kubernetes Service, Aurora and RDS, DynamoDB, VPC, S3, and AWS Amplify. The main area shows a list of security groups under 'Recent connections'. One specific security group is selected: 'sg-0b0d10be63d13dd9d - student-record-system-ecs-sg'. The 'Details' section shows the security group name ('student-record-system-ecs-sg'), security group ID ('sg-0b0d10be63d13dd9d'), description ('Security group for ECS tasks'), VPC ID ('vpc-036824e0a8e209300'), owner ('216989132235'), inbound rules count ('1 Permission entry'), and outbound rules count ('1 Permission entry'). Below this, the 'Inbound rules (1)' section is expanded, showing one rule: 'sgr-01d3ad50ba5deb8ba' (Security group rule ID), 'HTTP' (Type), 'TCP' (Protocol), and port '80'. There are buttons for 'Manage tags' and 'Edit inbound rules'.

Step 2.3: RDS Security Group

```
export RDS_SG_ID=$(aws ec2 create-security-group \
--group-name "$PROJECT_NAME-rds-sg" \
--description "Security group for RDS database" \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=security-
group,Tags=[{Key=Name,Value=$PROJECT_NAME-rds-sg},
{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'GroupId' \
--output text)
```

The screenshot shows a terminal window with the following content:

```
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export RDS_SG_ID=$(aws ec2 create-security-group \
--group-name "$PROJECT_NAME-rds-sg" \
--description "Security group for RDS database" \
--vpc-id $VPC_ID \
--tag-specifications "ResourceType=security-
group,Tags=[{Key=Name,Value=$PROJECT_NAME-rds-sg},
{Key=Environment,Value=$ENVIRONMENT}]" \
--query 'GroupId' \
--output text)
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ # Add rule to allow MySQL traffic from ECS
aws ec2 authorize-security-group-ingress \
--group-id $RDS_SG_ID \
--protocol tcp \
--port 3306 \
--source-group $ECS_SG_ID

echo "RDS Security Group: $RDS_SG_ID"
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0ec591cb8ec0fa793",
      "GroupId": "sg-00edc371c44ce03ea",
      "GroupOwnerId": "216989132235",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 3306,
      "ToPort": 3306,
      "ReferencedGroupInfo": {
        "GroupId": "sg-0b0d10be63d13dd9d",
        "UserId": "216989132235"
      },
      "SecurityGroupRuleArn": "arn:aws:ec2:eu-central-1:216989132235:security-group-rule/sgr-0ec591cb8ec0fa793"
    }
  ]
}
```

Add rule to allow MySQL traffic from ECS

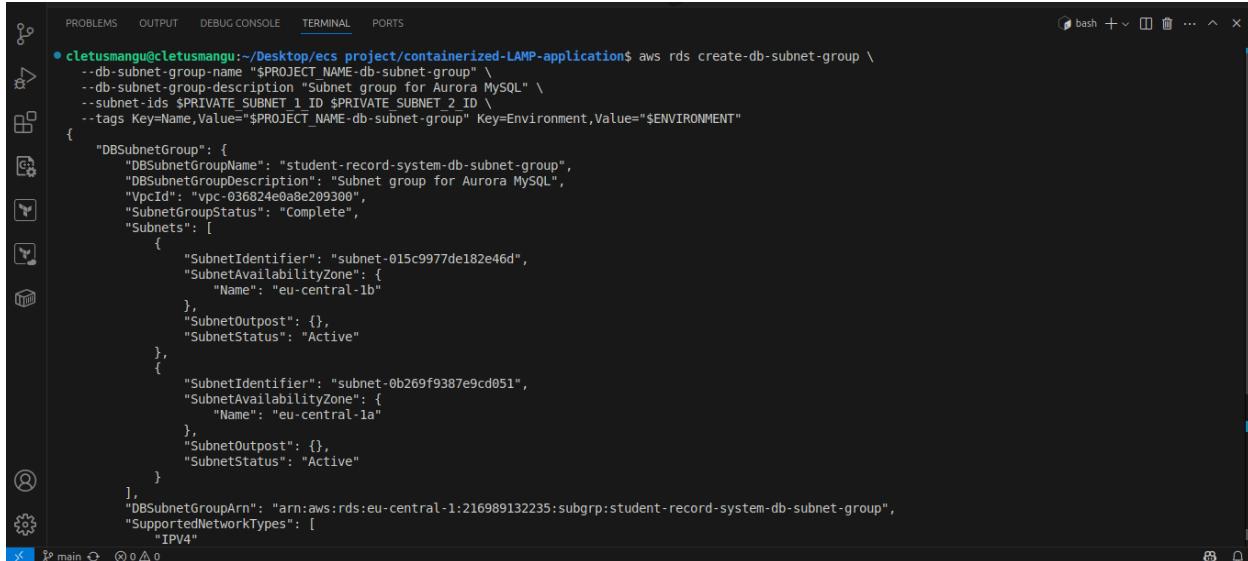
```
aws ec2 authorize-security-group-ingress \
--group-id $RDS_SG_ID \
--protocol tcp \
--port 3306 \
--source-group $ECS_SG_ID

echo "RDS Security Group: $RDS_SG_ID"
```

Phase 3: Create RDS Aurora MySQL Database

Step 3.1: Create DB Subnet Group

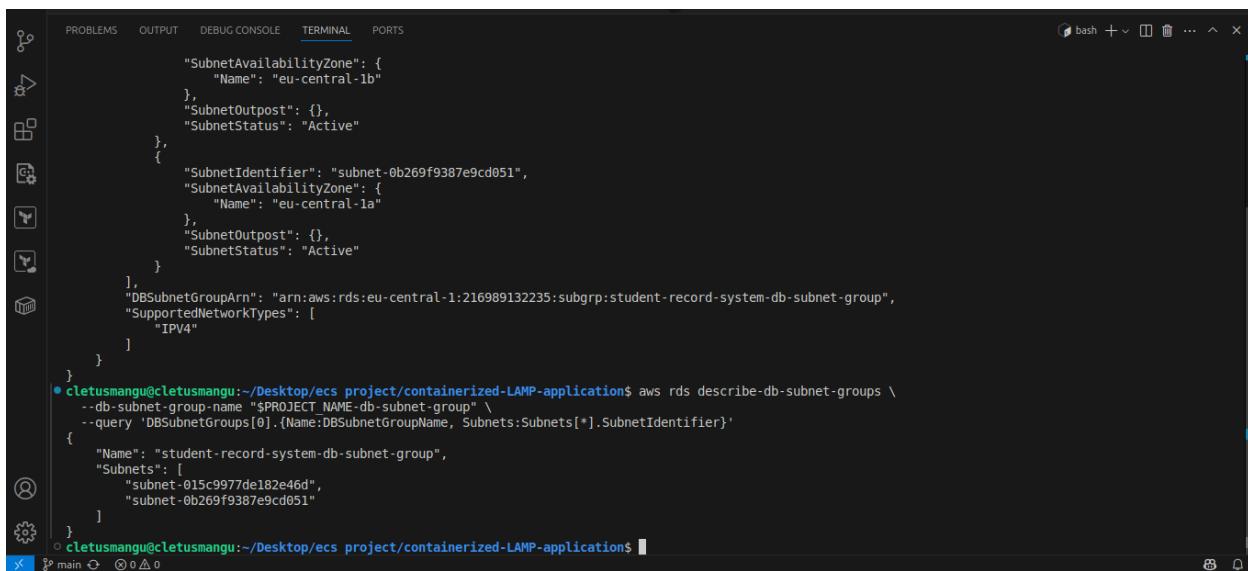
```
aws rds create-db-subnet-group \
--db-subnet-group-name "$PROJECT_NAME-db-subnet-group" \
--db-subnet-group-description "Subnet group for Aurora MySQL" \
--subnet-ids $PRIVATE_SUBNET_1_ID $PRIVATE_SUBNET_2_ID \
--tags Key=Name,Value="$PROJECT_NAME-db-subnet-group"
Key=Environment,Value="$ENVIRONMENT"
```



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws rds create-db-subnet-group \
--db-subnet-group-name "$PROJECT_NAME-db-subnet-group" \
--db-subnet-group-description "Subnet group for Aurora MySQL" \
--subnet-ids $PRIVATE_SUBNET_1_ID $PRIVATE_SUBNET_2_ID \
--tags Key=Name,Value="$PROJECT_NAME-db-subnet-group" Key=Environment,Value="$ENVIRONMENT"

{
  "DBSubnetGroup": {
    "DBSubnetGroupName": "student-record-system-db-subnet-group",
    "DBSubnetGroupDescription": "Subnet group for Aurora MySQL",
    "VpcId": "vpc-036824e0a8e209300",
    "SubnetGroupStatus": "Complete",
    "Subnets": [
      {
        "SubnetIdentifier": "subnet-015c9977de182e46d",
        "SubnetAvailabilityZone": {
          "Name": "eu-central-1b"
        },
        "SubnetOutpost": {},
        "SubnetStatus": "Active"
      },
      {
        "SubnetIdentifier": "subnet-0b269f9387e9cd051",
        "SubnetAvailabilityZone": {
          "Name": "eu-central-1a"
        },
        "SubnetOutpost": {},
        "SubnetStatus": "Active"
      }
    ],
    "DBSubnetGroupArn": "arn:aws:rds:eu-central-1:216989132235:subgrp:student-record-system-db-subnet-group",
    "SupportedNetworkTypes": [
      "IPV4"
    ]
  }
}
```

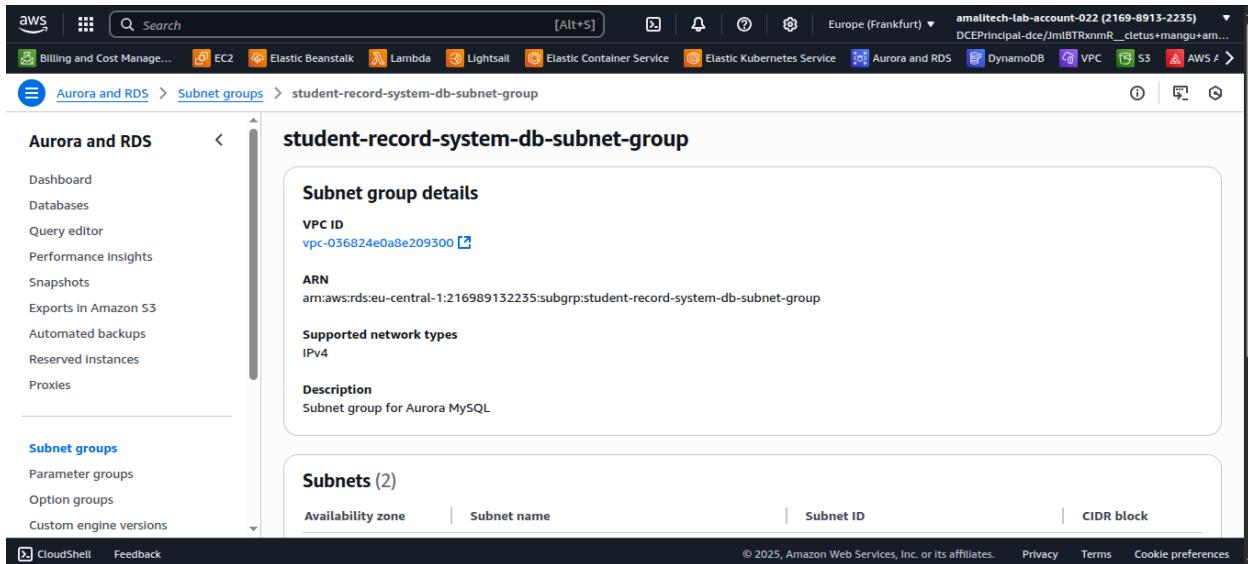


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
"SubnetAvailabilityZone": {
  "Name": "eu-central-1b"
},
"SubnetOutpost": {},
"SubnetStatus": "Active"
},
{
  "SubnetIdentifier": "subnet-0b269f9387e9cd051",
  "SubnetAvailabilityZone": {
    "Name": "eu-central-1a"
  },
  "SubnetOutpost": {},
  "SubnetStatus": "Active"
}
],
"DBSubnetGroupArn": "arn:aws:rds:eu-central-1:216989132235:subgrp:student-record-system-db-subnet-group",
"SupportedNetworkTypes": [
  "IPV4"
]
}

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws rds describe-db-subnet-groups \
--db-subnet-group-name "$PROJECT_NAME-db-subnet-group" \
--query 'DBSubnetGroups[0].{Name:DBSubnetGroupName, Subnets:Subnets[*].SubnetIdentifier}'

{
  "Name": "student-record-system-db-subnet-group",
  "Subnets": [
    "subnet-015c9977de182e46d",
    "subnet-0b269f9387e9cd051"
  ]
}
```



Step 3.2: Create Aurora MySQL Cluster

```

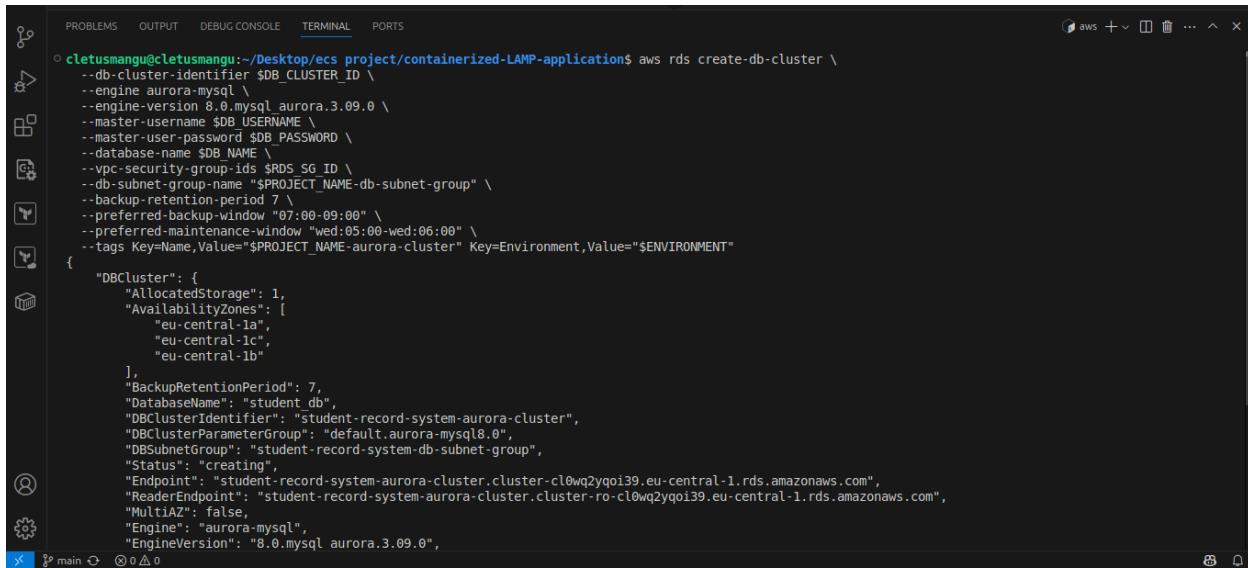
export DB_CLUSTER_ID="$PROJECT_NAME-aurora-cluster" aws rds create-db-cluster \
--db-cluster-identifier $DB_CLUSTER_ID \
--engine aurora-mysql \
--engine-version 8.0.mysql_aurora.3.02.0 \
--master-username $DB_USERNAME \
--master-user-password $DB_PASSWORD \
--database-name $DB_NAME \
--vpc-security-group-ids $RDS_SG_ID \
--db-subnet-group-name "$PROJECT_NAME-db-subnet-group" \
--backup-retention-period 7 \
--preferred-backup-window "07:00-09:00" \
--preferred-maintenance-window "wed:05:00-wed:06:00" \
--tags Key=Name,Value="$PROJECT_NAME-aurora-cluster" \
Key=Environment,Value="$ENVIRONMENT"

# Wait for cluster to be available

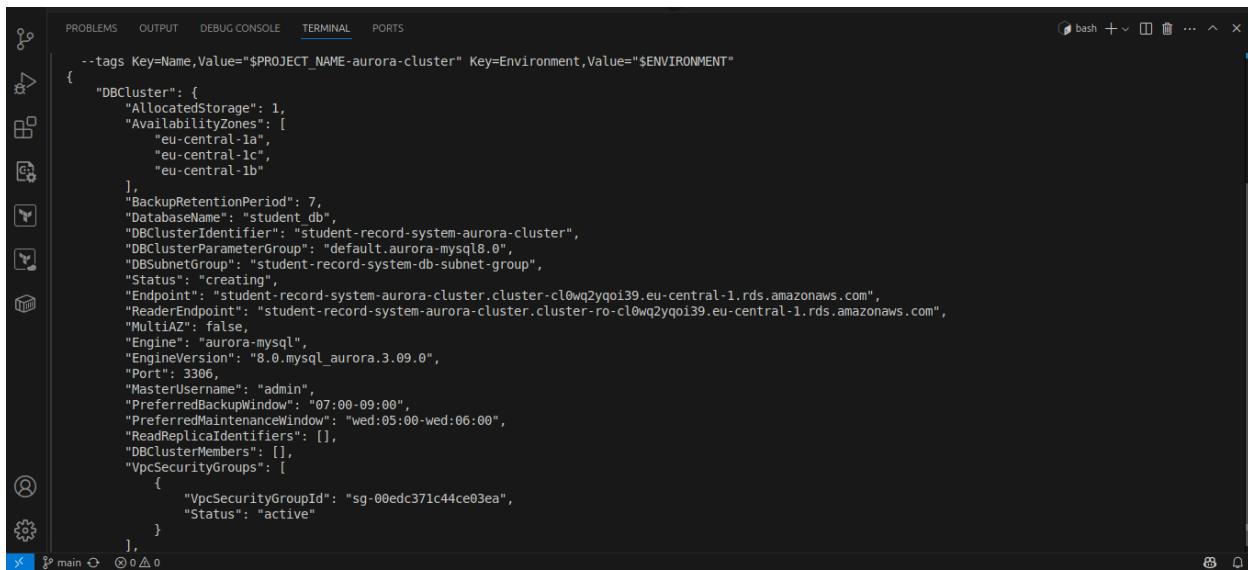
aws rds wait db-cluster-available

```

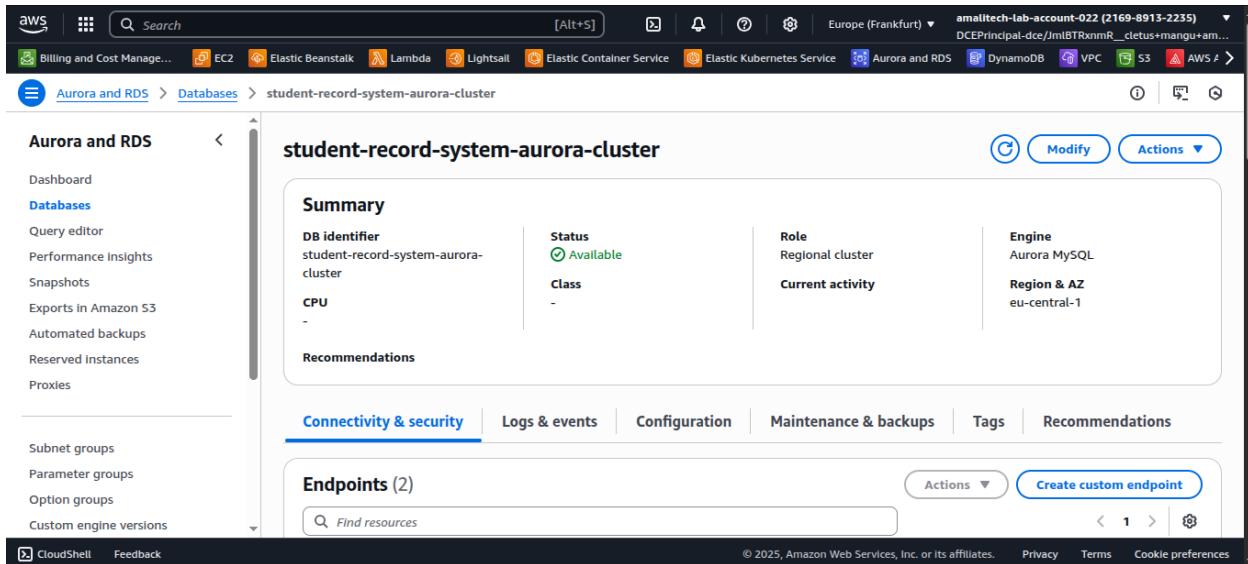
--db-cluster-identifier \$DB_CLUSTER_ID



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
aws rds create-db-cluster \
--db-cluster-identifier $DB_CLUSTER_ID \
--engine aurora-mysql \
--engine-version 8.0.mysql_aurora.3.09.0 \
--master-username $DB_USERNAME \
--master-user-password $DB_PASSWORD \
--database-name $DB_NAME \
--vpc-security-group-ids $RDS_SG_ID \
--db-subnet-group-name "$PROJECT_NAME-db-subnet-group" \
--backup-retention-period 7 \
--preferred-backup-window "07:00-09:00" \
--preferred-maintenance-window "wed:05:00-wed:06:00" \
--tags Key=Name,Value="$PROJECT_NAME-aurora-cluster" Key=Environment,Value="$ENVIRONMENT"
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "eu-central-1a",
            "eu-central-1c",
            "eu-central-1b"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "student_db",
        "DBClusterIdentifier": "student-record-system-aurora-cluster",
        "DBClusterParameterGroup": "default.aurora-mysql8.0",
        "DBSubnetGroup": "student-record-system-db-subnet-group",
        "Status": "creating",
        "Endpoint": "student-record-system-aurora-cluster.cluster-cl0wq2yqoi39.eu-central-1.rds.amazonaws.com",
        "ReaderEndpoint": "student-record-system-aurora-cluster.cluster-ro-cl0wq2yqoi39.eu-central-1.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-mysql",
        "EngineVersion": "8.0.mysql_aurora.3.09.0",
    }
}
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
--tags Key=Name,Value="$PROJECT_NAME-aurora-cluster" Key=Environment,Value="$ENVIRONMENT"
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "eu-central-1a",
            "eu-central-1c",
            "eu-central-1b"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "student_db",
        "DBClusterIdentifier": "student-record-system-aurora-cluster",
        "DBClusterParameterGroup": "default.aurora-mysql8.0",
        "DBSubnetGroup": "student-record-system-db-subnet-group",
        "Status": "creating",
        "Endpoint": "student-record-system-aurora-cluster.cluster-cl0wq2yqoi39.eu-central-1.rds.amazonaws.com",
        "ReaderEndpoint": "student-record-system-aurora-cluster.cluster-ro-cl0wq2yqoi39.eu-central-1.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-mysql",
        "EngineVersion": "8.0.mysql_aurora.3.09.0",
        "Port": 3306,
        "MasterUsername": "admin",
        "PreferredBackupWindow": "07:00-09:00",
        "PreferredMaintenanceWindow": "wed:05:00-wed:06:00",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-00edc371c44ce03ea",
                "Status": "active"
            }
        ],
    }
}
```



Step 3.3: Create Aurora MySQL Instance

```
aws rds create-db-instance \
--db-instance-identifier "$PROJECT_NAME-aurora-instance-1" \
--db-instance-class db.t3.medium \
--engine aurora-mysql \
--db-cluster-identifier $DB_CLUSTER_ID \
--tags Key=Name,Value="$PROJECT_NAME-aurora-instance-1" \
Key=Environment,Value="$ENVIRONMENT"

# Get DB endpoint

export DB_ENDPOINT=$(aws rds describe-db-clusters \
--db-cluster-identifier $DB_CLUSTER_ID \
--query 'DBClusters[0].Endpoint' \
--output text)

echo "Database Endpoint: $DB_ENDPOINT"
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws rds create-db-instance \
--db-instance-identifier "$PROJECT_NAME-aurora-instance-1" \
--db-instance-class db.t3.medium \
--engine aurora-mysql \
--db-cluster-identifier $DB_CLUSTER_ID \
--tags Key=Name,Value="$PROJECT_NAME-aurora-instance-1" Key=Environment,Value="$ENVIRONMENT"
{
  "DBInstance": {
    "DBInstanceIdentifier": "student-record-system-aurora-instance-1",
    "DBInstanceClass": "db.t3.medium",
    "Engine": "aurora-mysql",
    "DBInstanceStatus": "creating",
    "MasterUsername": "admin",
    "DBName": "student_db",
    "AllocatedStorage": 1,
    "PreferredBackupWindow": "07:00-09:00",
    "BackupRetentionPeriod": 7,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
      {
        "VpcSecurityGroupId": "sg-00edc371c44ce03ea",
        "Status": "active"
      }
    ],
    "DBParameterGroups": [
      {
        "DBParameterGroupName": "default.aurora-mysql8.0",
        "ParameterApplyStatus": "in-sync"
      }
    ],
    "DBSubnetGroup": {
      "DBSubnetGroupName": "student-record-system-db-subnet-group",
      "DBSubnetGroupDescription": "Aurora DB Subnet Group for student-record-system-aurora-instance-1"
    }
  }
}

```

Aurora and RDS > Databases > student-record-system-aurora-cluster > student-record-system-aurora-instance-1

DB identifier	Status	Role	Engine	Region	Size	Rec...
student-record-system-aurora-cluster	Available	Regional c...	Aurora My...	eu-central-1	1 Instance	
student-record-system-aurora-instance-1	Available	Writer ins...	Aurora My...	eu-central...	db.t3.med...	

Connectivity & security

Phase 4: Create ECR Repository and Build Docker Image

Step 4.1: Create ECR Repository

```

aws ecr create-repository \
--repository-name $PROJECT_NAME \
--image-scanning-configuration scanOnPush=true \

```

```
--tags Key=Name,Value="$PROJECT_NAME-ecr"
Key=Environment,Value="$ENVIRONMENT"
```

Export

```
ECR_REPOSITORY_URI="$AWS_ACCOUNT_ID.dkr.ecr.$AWS_REGION.amazonaws.com/
$PROJECT_NAME"
```

```
echo "ECR Repository: $ECR_REPOSITORY_URI"
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ aws ecr create-repository \
--repository-name $PROJECT_NAME \
--image-scanning-configuration scanOnPush=true \
--tags Key=Name,Value="$PROJECT_NAME-ecr" Key=Environment,Value="$ENVIRONMENT"
{
  "repository": {
    "repositoryArn": "arn:aws:ecr:eu-central-1:216989132235:repository/student-record-system",
    "registryId": "216989132235",
    "repositoryName": "student-record-system",
    "repositoryUri": "216989132235.dkr.ecr.eu-central-1.amazonaws.com/student-record-system",
    "createdAt": "2025-07-05T0:12:24.540000+00:00",
    "imageTagMutability": "MUTABLE",
    "imageScanningConfiguration": {
      "scanOnPush": true
    },
    "encryptionConfiguration": {
      "encryptionType": "AES256"
    }
  }
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$ export ECR_REPOSITORY_URI="$AWS_ACCOUNT_ID.dkr.ecr.$AWS_REGION.amazonaws.com/$PROJECT_NAME"
echo "ECR Repository: $ECR_REPOSITORY_URI"
ECR Repository: 216989132235.dkr.ecr.eu-central-1.amazonaws.com/student-record-system
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application$
```

Private repositories (1)

Repository name	URI	Created at	Tag immutability	Encryption type
student-record-system	216989132235.dkr.ecr.eu-central-1.amazonaws.com/student-record-system	July 05, 2025, 10:12:24 (UTC-00)	Mutable	AES-256

Step 4.2: Build and Push Docker Image

```
# Create Dockerfile (save as Dockerfile)
cat > Dockerfile << EOF
```

```
FROM php:8.1-apache

# Install system dependencies
RUN apt-get update && apt-get install -y \
    git \
    curl \
    libpng-dev \
    libonig-dev \
    libxml2-dev \
    zip \
    unzip \
    default-mysql-client \
    && rm -rf /var/lib/apt/lists/*

# Install PHP extensions
RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd

# Enable Apache mod_rewrite
RUN a2enmod rewrite

# Set working directory
WORKDIR /var/www/html

# Copy application source code
COPY src/ /var/www/html/

# Set proper permissions
RUN chown -R www-data:www-data /var/www/html \
    && chmod -R 755 /var/www/html

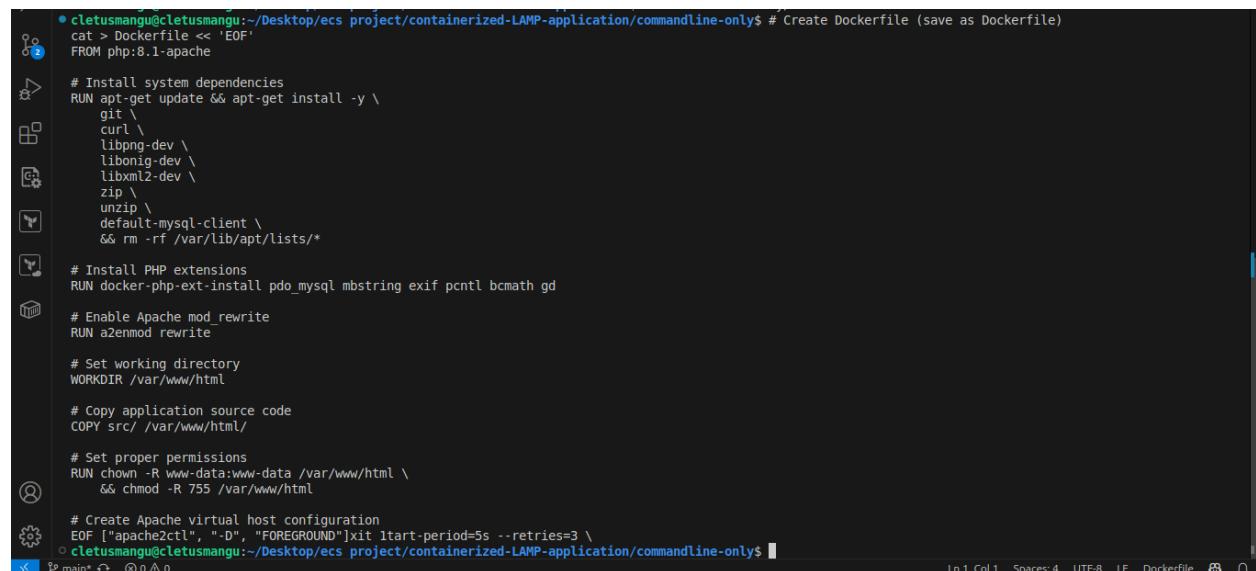
# Create Apache virtual host configuration
RUN echo '<VirtualHost *:80>\n' \
    DocumentRoot /var/www/html\n' \
    ServerName localhost\n' \
    <Directory "/var/www/html">\n' \
        AllowOverride All\n' \
        Require all granted\n' \
    </Directory>\n' \
    ErrorLog ${APACHE_LOG_DIR}/error.log\n' \
    CustomLog ${APACHE_LOG_DIR}/access.log combined\n'
```

```
</VirtualHost>' > /etc/apache2/sites-available/000-default.conf
```

```
# Expose port 80
EXPOSE 80

# Health check
HEALTHCHECK --interval=30s --timeout=3s --start-period=5s --retries=3 \
CMD curl -f http://localhost/ || exit 1
```

```
# Start Apache in foreground
CMD ["apache2ctl", "-D", "FOREGROUND"]
EOF
```



The screenshot shows a terminal window with a dark background and light-colored text. It displays a Dockerfile with various commands for setting up a LAMP application. The file includes commands for installing dependencies like git, curl, libpng-dev, libonig-dev, libxml2-dev, zip, and unzip. It also installs PHP extensions (pdo_mysql, mbstring, exif, pcntl, bcmath, gd), enables mod_rewrite, sets the working directory to /var/www/html, copies application source code, and sets proper permissions. Finally, it creates an Apache virtual host configuration and starts the server.

```
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ ``Create Dockerfile (save as Dockerfile)
cat > Dockerfile << 'EOF'
FROM php:8.1-apache

# Install system dependencies
RUN apt-get update && apt-get install -y \
    git \
    curl \
    libpng-dev \
    libonig-dev \
    libxml2-dev \
    zip \
    unzip \
    default-mysql-client \
    && rm -rf /var/lib/apt/lists/*

# Install PHP extensions
RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd

# Enable Apache mod_rewrite
RUN a2enmod rewrite

# Set working directory
WORKDIR /var/www/html

# Copy application source code
COPY src/ /var/www/html/

# Set proper permissions
RUN chown -R www-data:www-data /var/www/html \
    && chmod -R 755 /var/www/html

# Create Apache virtual host configuration
EOF "["apache2ctl", "-D", "FOREGROUND"]xit 1start-period=5s --retries=3 \
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

```
# Create application files directory
mkdir -p src/assets
```

```
# Create config.php
cat > src/config.php << 'EOF'
<?php
$servername = $_ENV['DB_HOST'] ?? 'localhost';
$username = $_ENV['DB_USER'] ?? 'root';
$password = $_ENV['DB_PASSWORD'] ?? '';
$dbname = $_ENV['DB_NAME'] ?? 'student_db';
$port = $_ENV['DB_PORT'] ?? 3306;
```

```
try {
```

```

$pdo = new PDO("mysql:host=$servername;port=$port;dbname=$dbname", $username,
$password);
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// Create students table if it doesn't exist
$sql = "CREATE TABLE IF NOT EXISTS students (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    age INT(3) NOT NULL,
    department VARCHAR(100) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)";
$pdo->exec($sql);

} catch(PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}
?>
EOF

```

The screenshot shows a terminal window with the following content:

```

● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create application files directory
mkdir -p src/assets

# Create config.php
cat > src/config.php << 'EOF'
<?php
$servername = $_ENV['DB_HOST'] ?? 'localhost';
$username = $_ENV['DB_USER'] ?? 'root';
$password = $_ENV['DB_PASSWORD'] ?? '';
$dbname = $_ENV['DB_NAME'] ?? 'student_db';
$port = $_ENV['DB_PORT'] ?? 3306;

try {
    $pdo = new PDO("mysql:host=$servername;port=$port;dbname=$dbname", $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Create students table if it doesn't exist
    $sql = "CREATE TABLE IF NOT EXISTS students (
        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        name VARCHAR(100) NOT NULL,
        age INT(3) NOT NULL,
        department VARCHAR(100) NOT NULL,
        created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
    )";
    $pdo->exec($sql);

} catch(PDOException $e) {
    die("Connection failed: " . $e->getMessage());
}
?>
EOF
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ 

```

The terminal shows the command to create a directory, the creation of a config.php file, and the execution of the PHP code which creates a students table in a MySQL database.

```

# Create index.php (simplified version)
cat > src/index.php << 'EOF'
<?php
require_once 'config.php';

```

```

// Fetch all students
try {
    $stmt = $pdo->query("SELECT * FROM students ORDER BY created_at DESC");
    $students = $stmt->fetchAll(PDO::FETCH_ASSOC);
} catch(PDOException $e) {
    $students = [];
    $error_message = "Error fetching students: " . $e->getMessage();
}

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['action'])) {
    if ($_POST['action'] == 'add' && !empty($_POST['name']) && !empty($_POST['age'])
&& !empty($_POST['department'])) {
        try {
            $stmt = $pdo->prepare("INSERT INTO students (name, age, department) VALUES
(?, ?, ?)");
            $stmt->execute([$_.POST['name'], (int)$_.POST['age'], $_POST['department']]);
            header("Location: " . $_SERVER['PHP_SELF']);
            exit();
        } catch(PDOException $e) {
            $error_message = "Error adding student: " . $e->getMessage();
        }
    } elseif ($_POST['action'] == 'delete' && !empty($_POST['id'])) {
        try {
            $stmt = $pdo->prepare("DELETE FROM students WHERE id = ?");
            $stmt->execute([(int)$_POST['id']]);
            header("Location: " . $_SERVER['PHP_SELF']);
            exit();
        } catch(PDOException $e) {
            $error_message = "Error deleting student: " . $e->getMessage();
        }
    }
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<title>Student Record System</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
<style>
  body { background-color: #f8f9fa; }
  .header-section {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white; padding: 3rem 2rem; border-radius: 10px; margin-bottom: 2rem;
  }
  .table { background-color: white; border-radius: 10px; overflow: hidden; }
</style>
</head>
<body>
  <div class="container mt-5">
    <div class="header-section text-center">
      <h1 class="display-4">Student Record System</h1>
      <p class="lead">Containerized LAMP Application on AWS ECS Fargate</p>
    </div>

    <?php if (isset($error_message)): ?>
      <div class="alert alert-danger"><?php echo htmlspecialchars($error_message); ?></div>
    <?php endif; ?>

    <!-- Add Student Form -->
    <div class="card mb-4">
      <div class="card-header">
        <h3>Add New Student</h3>
      </div>
      <div class="card-body">
        <form method="POST">
          <input type="hidden" name="action" value="add">
          <div class="row">
            <div class="col-md-4">
              <input type="text" class="form-control" name="name" placeholder="Full
Name" required>
            </div>
            <div class="col-md-2">
              <input type="number" class="form-control" name="age" placeholder="Age"
min="16" max="100" required>
            </div>

```

```

<div class="col-md-4">
    <select class="form-control" name="department" required>
        <option value="">Select Department</option>
        <option value="Computer Science">Computer Science</option>
        <option value="Engineering">Engineering</option>
        <option value="Business">Business</option>
        <option value="Medicine">Medicine</option>
        <option value="Arts">Arts</option>
    </select>
</div>
<div class="col-md-2">
    <button type="submit" class="btn btn-primary w-100">Add Student</button>
</div>
</div>
</form>
</div>
</div>

<!-- Students Table -->
<div class="card">
    <div class="card-header">
        <h3>All Students (<?php echo count($students); ?> total)</h3>
    </div>
    <div class="card-body">
        <div class="table-responsive">
            <table class="table table-striped">
                <thead class="table-dark">
                    <tr>
                        <th>ID</th>
                        <th>Name</th>
                        <th>Age</th>
                        <th>Department</th>
                        <th>Added On</th>
                        <th>Actions</th>
                    </tr>
                </thead>
                <tbody>
                    <?php if (empty($students)): ?>
                    <tr>
                        <td colspan="6" class="text-center">No students found. Add your first

```

```

student!</td>
    </tr>
<?php else: ?>
    <?php foreach ($students as $student): ?>
        <tr>
            <td><?php echo htmlspecialchars($student['id']); ?></td>
            <td><?php echo htmlspecialchars($student['name']); ?></td>
            <td><?php echo htmlspecialchars($student['age']); ?></td>
            <td><?php echo htmlspecialchars($student['department']); ?></td>
            <td><?php echo date('M d, Y',
strtotime($student['created_at'])); ?></td>
            <td>
                <form method="POST" style="display:inline;">
                    <input type="hidden" name="action" value="delete">
                    <input type="hidden" name="id" value="<?php echo
$student['id']; ?>">
                    <button type="submit" class="btn btn-danger btn-sm"
                        onclick="return confirm('Are you sure?')">Delete</button>
                </form>
            </td>
        </tr>
    <?php endforeach; ?>
    <?php endif; ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
EOF

```

```

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create index.php (simplified version)
cat > src/index.php << 'EOF'
<?php
require_once 'config.php';

// Fetch all students
try {
    $stmt = $pdo->query("SELECT * FROM students ORDER BY created_at DESC");
    $students = $stmt->fetchAll(PDO::FETCH_ASSOC);
} catch(PDOException $e) {
    $students = [];
    $error_message = "Error fetching students: " . $e->getMessage();
}

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['action'])) {
    if ($_POST['action'] == 'add' && !empty($_POST['name']) && !empty($_POST['age']) && !empty($_POST['department'])) {
        try {
            $stmt = $pdo->prepare("INSERT INTO students (name, age, department) VALUES (?, ?, ?)");
            $stmt->execute([$_POST['name'], (int)$_POST['age'], $_POST['department']]);
            header("Location: " . $_SERVER['PHP_SELF']);
            exit();
        } catch(PDOException $e) {
            $error_message = "Error adding student: " . $e->getMessage();
        }
    } elseif ($_POST['action'] == 'delete' && !empty($_POST['id'])) {
        try {
            $stmt = $pdo->prepare("DELETE FROM students WHERE id = ?");
            $stmt->execute([(int)$_POST['id']]);
            header("Location: " . $_SERVER['PHP_SELF']);
            exit();
        } catch(PDOException $e) {
            $error_message = "Error deleting student: " . $e->getMessage();
        }
    }
}
EOF

```

html src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script></button>?">d>quired>

Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | Dockerfile |

Login to ECR

```
aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REPOSITORY_URI
```

```

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REPOSITORY_URI
WARNING! Your credentials are stored unencrypted in '/home/cletusmangu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

```

Login Succeeded

```

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Build and push image
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ docker build -t $PROJECT_NAME .
[+] Building 5.8s (5/13)
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/php:8.1-apache
=> [auth] library/php:pull token for registry-1.docker.io
=> [internal] load .dockerrcignore
=> [internal] load context: 2B
=> [internal] load build context
=> [internal] load context: 7.82kB
=> [1/8] FROM docker.io/library/php:8.1-apache@sha256:f1b67b9ac4a103c6cde1a1eb3b5e06a5b4e851ee8858c763fe3b122d82768a13
=> resolve docker.io/library/php:8.1-apache@sha256:f1b67b9ac4a103c6cde1a1eb3b5e06a5b4e851ee8858c763fe3b122d82768a13
=> sha256:5aea8e4e81017a0a03499beeb569ba80aa259a7f66199612b2b9f61366c4d08 11.61kB / 11.61kB
=> sha256:f1b67b9ac4a103c6cde1a1eb3b5e06a5b4e851ee8858c763fe3b122d82768a13 10.40kB / 10.40kB
=> sha256:3da95a905ed546f99c4564407923a68175d89651a388ec3ff5e9bf5ed0b39d 0B / 28.23MB
=> sha256:cdd80672db45bd21268b018ee45930631ab31aafe2eaba6665c7e262b98a014 227B / 227B
=> sha256:73ae33c3742666c63fcbae0424b15bc0a183f35bbe66ca67bb5672d179132768f8a2cd 0B / 104.33MB
=> sha256:9eb567fb626e398352ebaeb3117cc5cle6d790ea17cf05488e4241c0a20947 0B / 225B

```

Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | Dockerfile |

Build and push image

```
docker build -t $PROJECT_NAME .
```

```
docker tag $PROJECT_NAME:latest $ECR_REPOSITORY_URI:latest
```

```
docker push $ECR_REPOSITORY_URI:latest
```

```
echo "Docker image pushed to: $ECR_REPOSITORY_URI:latest"
```

```

cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws ecr get-login-password --region $AWS_REGION | docker login --username AWS --password-stdin $ECR_REPOSITORY_URI
WARNING! Your credentials are stored unencrypted in '/home/cletusmangu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Succeeded
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Build and push image
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ docker build -t $PROJECT_NAME .
[+] Building 5.8s (5/13)
=> [internal] Load build definition from Dockerfile
=> [internal] Load build context
=> [internal] Load metadata for docker.io/library/php:8.1-apache
=> [auth] library/php:pull token for registry-1.docker.io
=> [internal] Load .dockignore
=> [internal] Transfer context: 2B
=> [internal] Load build context
=> [internal] Transfer context: 7.8kB
=> [1/8] FROM docker.io/library/php:8.1-apache@sha256:f1b67b9ac4a103c6cde1a1eb3b5e06a5b4e851ee8858c763fe3b122d82768a13
=> => resolve docker.io/library/php:8.1-apache@sha256:f1b67b9ac4a103c6cde1a1eb3b5e06a5b4e851ee8858c763fe3b122d82768a13
=> => sha256:5aea8e4e81617a069b8a0259a7f766190612b29f6f1360e64d8 11.61kB / 11.61kB
=> => sha256:f1b67b9ac4a103c6cde1a1eb3b5e06a5b4e851ee8858c763fe3b122d82768a13 1.55
=> => sha256:3da95a905ed5461f99c4564407923a68175d89951a8f5ed0b39d 0B / 28.23MB
=> => sha256:3f99d7cb2bae042ba415bc0a183f35b6b6ca67bb5672d179132768f8a2cd 0B / 104.33MB
=> => sha256:cdd80672db45bd21268b018ee45939631ab31aafe2e3a6665c7e262b98a014 227B / 227B
=> => sha256:73ae33c3742666c63fc7530a9aa2644eee0c475c32b94391b360cb7700aa8f9 3.83kB / 3.83kB
=> => sha256:9eb567fb626e398352ebaeb3117cc5c1e6d7900ea17cf05488e4241c0a20947 0B / 225B
=> => sha256:b53313722a28a37fcf362c544cf3fd7d373aeca9208516d23db137f1716f2f6595 431B / 431B
=> => sha256:1660f79bd2c2fc7e54452c43d168e8c5be69a64e21a9f595a56fc2d6e4c3f310c 486B / 486B
=> => sha256:41f34298e3d069fb6091d421046702726fe5cf614ced01369e1e9d5c6c03e7b 12.03MB / 12.03MB
=> => sha256:bfa41226235b78d5e12ca7de5e5807e3f407df479930ec41d3d45c00e2edbfff4 499B / 499B
=> => sha256:88cd14855df225513db04a9c93aae63acc43bcc995b2df5874d30f259f4c6dd 11.16MB / 11.16MB
=> => extracting sha256:3da95a905ed546f99c4564407923a68175d89951a8f5ed0b39d 25.25
=> => sha256:225778fdeccf587adc6c01e79895e290ba7e44c867087cd3798cf92cab360e6bd3 2.46kB / 2.46kB
=> => sha256:89476a2bbbd0cee95c6573267404434d22b06b23c353a161d7769dcda18cc 248B / 248B
=> => extracting sha256:cdd80672db45bd21268b018ee45939631ab31aafe2e3a6665c7e262b98a014 59.05
=> => sha256:1215c71937e9f773a96e099bc796d591445f2857dicac1c752ddaa8c89814c654 243B / 243B
=> => sha256:fed3465857859935feb9715f96f087741a2475c0475757e905d1484f 892B / 892B
=> => sha256:4f4fb700ef54461cfa02571ae0db9a0d1c0e6bd5577484ade75e68d38e8aacc1 32B / 32B
=> => extracting sha256:3f90d7cb2bae042ba415bc0a183f35b6b6ca67bb5672d179132768f8a2cd 2.35
=> => extracting sha256:9eb567fb626e398352ebaeb3117cc5c1e6d7900ea17cf05488e4241c0a20947 0.05
=> => extracting sha256:a041ea15491fc46539cf8448b5b9a32599fa66837da558bb2b2a05d1475b370 0.35
=> => extracting sha256:b53313722a28a37fcf362c544cf3fd7d373aeca9208516d23db137f1716f2f6595 0.05
=> => extracting sha256:1660f79bd2c2fc7e54452c43d168e8c5be69a6421a9f595a56fc2d6e4c3f310c 0.05
=> => extracting sha256:41f34298e3d069fb6091d421046702726fe5cf614ced01369e1e9d5c6c03e7b 0.05
=> => extracting sha256:bfa41262350f805e12ca7de5e5807e3f407df479930ec41d3d45c00e2edbfff4 0.05
=> => extracting sha256:88cd14855df225f3d0b4a9c93aae63acc43bcc995b2df5874d30f259f4c6dd 0.35
=> => extracting sha256:56573267404434d22b06b23c353a161d7769dcda18cc 0.05
=> => extracting sha256:89476a2bbbd0cee95c6573267404434d22b06b23c353a161d7769dcda18cc 0.05
=> => extracting sha256:1215c71937e9f773a96e099bc796d591445f2857dicac1c752ddaa8c89814c654 0.05
=> => extracting sha256:fed3465857859935feb9715f96f087741a6371e2a475c0475757e905d1484f 0.05
=> => extracting sha256:4f4fb700ef54461cfa02571ae0db9a0d1c0e6bd5577484ae675e68d38e8aacc1 0.05
=> [2/8] RUN apt-get update && apt-get install -y git curl libpng-dev libonig-dev libxml2-dev zip unzip default-mysql-client
=> [3/8] RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd
=> [4/8] RUN azemmod rewrite
=> [5/8] WORKDIR /var/www/html
=> [6/8] COPY src /var/www/html
=> [7/8] RUN chown -R www-data:www-data /var/www/html && chmod -R 755 /var/www/html
=> [8/8] RUN echo <VirtualHost *:80>\n    DocumentRoot /var/www/html\n    ServerName localhost\n    <Directory "/var/www/html">\n        AllowOverride All\n    </Directory>\n</VirtualHost>
=> exporting to image
=> exporting layers
=> writing image sha256:9f39bf4e3c21708e282766b24d32d99f3ea93278e172bf52e3631705f28070f9
=> naming to docker.io/library/student-record-system
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

```
  => [3/8] RUN docker-php-ext-install pdo_mysql mbstring exif pcntl bcmath gd
  => [4/8] RUN azemmod rewrite
  => [5/8] WORKDIR /var/www/html
  => [6/8] COPY src/ /var/www/html/
  => [7/8] RUN chown -R www-data:www-data /var/www/html    && chmod -R 755 /var/www/html
  => [8/8] RUN echo '<VirtualHost *:80>\n  DocumentRoot /var/www/html\n  ServerName localhost\n  <Directory "/var/www/html">\n    AllowOverride All\n</Directory>\n</VirtualHost>' | tee /etc/httpd/conf.d/default.conf
  => exporting to image
  => exporting layers
  => writing image sha256:9f39bf4e3c21708e282766b24d32d99f3ea93278e172bf52e3631705f28070f9
  => naming to docker.io/library/student-record-system
● cletusmangu@Letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ docker tag $PROJECT_NAME:latest $ECR_REPOSITORY_URI:latest
● cletusmangu@Letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ docker push $ECR_REPOSITORY_URI:latest

echo "Docker image pushed to: $ECR REPOSITORY URI:latest"
The push refers to repository [216989132235.dkr.ecr.eu-central-1.amazonaws.com/student-record-system]
55a9c11b1f2d: Pushed
fc5533e37d06: Pushed
51c9a433a80c: Pushed
5f70bf18a086: Pushed
ed99a3f73ce6: Pushed
4b7b92992ca0: Pushed
fe440b55610: Pushing [=====] 109.4MB/182.1MB
56e26342e587: Pushed
2eb0e9b848b7: Pushed
cc4de6ccdf2a: Pushed
8da4bba18a8c: Pushed
2e72be518421: Pushing [=====] 33.99MB/48.58MB
03d0f425ce4b: Pushed
2b07f17c0698: Pushing [=====] 4.861MB/12.39MB
467e5c3ebc78: Pushed
6c815fa1cc3c: Pushed
1a0153b9c84: Waiting
b72d37b5dde8: Waiting
441bf9b00d03: Waiting
6a2502656e90: Waiting
1bb35e8b4de1: Waiting

```

```
  => [5/8] WORKDIR /var/www/html
  => [6/8] COPY src/ /var/www/html/
  => [7/8] RUN chown -R www-data:www-data /var/www/html    && chmod -R 755 /var/www/html
  => [8/8] RUN echo '<VirtualHost *:80>\n  DocumentRoot /var/www/html\n  ServerName localhost\n  <Directory "/var/www/html">\n    AllowOverride All\n</Directory>\n</VirtualHost>' | tee /etc/httpd/conf.d/default.conf
  => exporting to image
  => exporting layers
  => writing image sha256:9f39bf4e3c21708e282766b24d32d99f3ea93278e172bf52e3631705f28070f9
  => naming to docker.io/library/student-record-system
● cletusmangu@Letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ docker tag $PROJECT_NAME:latest $ECR_REPOSITORY_URI:latest
● cletusmangu@Letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ docker push $ECR_REPOSITORY_URI:latest

echo "Docker image pushed to: $ECR REPOSITORY URI:latest"
The push refers to repository [216989132235.dkr.ecr.eu-central-1.amazonaws.com/student-record-system]
55a9c11b1f2d: Pushed
fc5533e37d06: Pushed
51c9a433a80c: Pushed
5f70bf18a086: Pushed
ed99a3f73ce6: Pushed
4b7b92992ca0: Pushed
fe440b55610: Pushed
56e26342e587: Pushed
2eb0e9b848b7: Pushed
cc4de6ccdf2a: Pushed
8da4bba18a8c: Pushed
2e72be518421: Pushed
03d0f425ce4b: Pushed
2b07f17c0698: Pushed
467e5c3ebc78: Pushed
6c815fa1cc3c: Pushed
1a0153b9c84: Pushed
b72d37b5dde8: Pushed
441bf9b00d03: Pushed
6a2502656e90: Pushed
1bb35e8b4de1: Pushed
latest: digest: sha256:ae2282430ca32e88979b1ac9e3974380b0b9e6e7alac27ab322b2c8420f8b55 size: 4907
Docker image pushed to: 216989132235.dkr.ecr.eu-central-1.amazonaws.com/student-record-system:latest
● cletusmangu@Letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

The screenshot shows the AWS Management Console with the ECR service selected. The left sidebar shows the navigation path: Amazon ECR > Private registry > Repositories > student-record-system. The main content area is titled 'Images (1)' and lists a single image named 'latest'. The table columns include Image tag, Artifact type, Pushed at, Size (MB), Image URI, Digest, and Last recorded pull time. The 'latest' image was pushed on July 05, 2025, at 10:40:10 (UTC-00), is 236.64 MB in size, and has a digest of sha256:aee2282430ca32... .

Phase 5: Create IAM Roles

Step 5.1: ECS Task Execution Role

```
# Create trust policy for ECS tasks
cat > ecs-task-trust-policy.json << 'EOF'
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

```
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create trust policy for ECS tasks
cat > ecs-task-trust-policy.json << 'EOF'
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create ECS Task Execution Role
aws iam create-role \
    --role-name "$PROJECT_NAME-ecs-task-execution-role" \
    --assume-role-policy-document file://ecs-task-trust-policy.json \
    --tags Key=Name,Value="$PROJECT_NAME-ecs-task-execution-role" Key=Environment,Value="$ENVIRONMENT"
{
    "Role": {
        "Path": "/",
        "RoleName": "student-record-system-ecs-task-execution-role",
        "RoleId": "AROATFBMPMXFV3WFLOXH5",
        "Arn": "arn:aws:iam::216989132235:role/student-record-system-ecs-task-execution-role",
        "CreateDate": "2025-07-05T04:42:42+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ecs-tasks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
Ln 1, Col 1  Spaces:4  UTF-8  LF  Dockerfile  ⌂  ⌂
```

Create ECS Task Execution Role

```
aws iam create-role \
    --role-name "$PROJECT_NAME-ecs-task-execution-role" \
    --assume-role-policy-document file://ecs-task-trust-policy.json \
    --tags Key=Name,Value="$PROJECT_NAME-ecs-task-execution-role" \
Key=Environment,Value="$ENVIRONMENT"
```

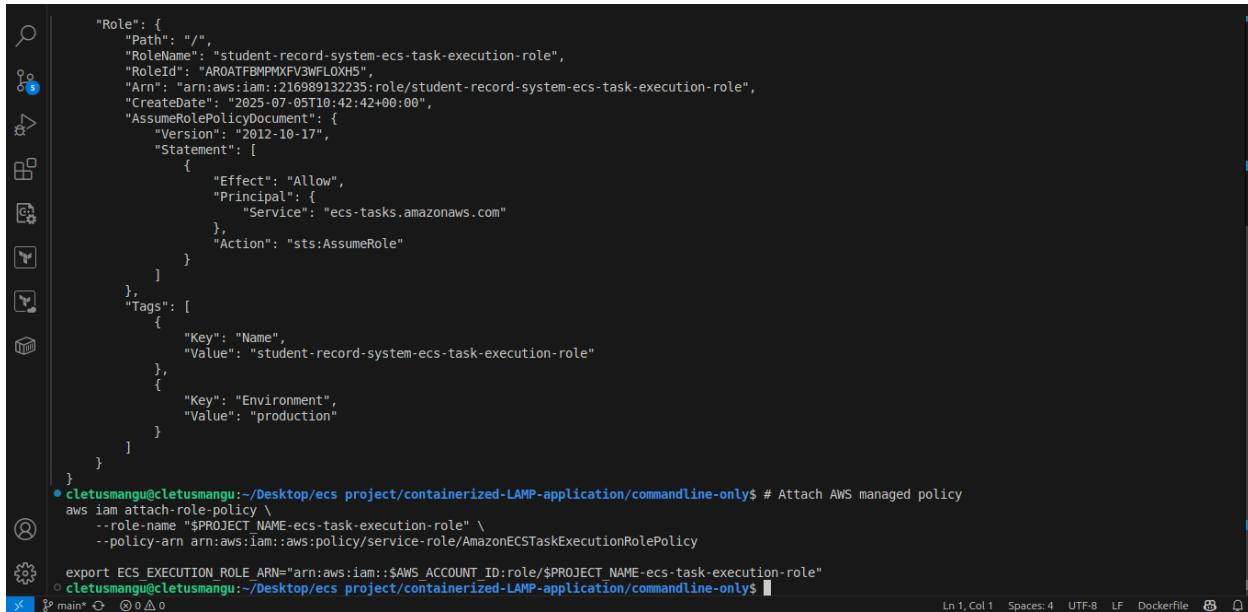
```
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create ECS Task Execution Role
aws iam create-role \
    --role-name "$PROJECT_NAME-ecs-task-execution-role" \
    --assume-role-policy-document file://ecs-task-trust-policy.json \
    --tags Key=Name,Value="$PROJECT_NAME-ecs-task-execution-role" Key=Environment,Value="$ENVIRONMENT"
{
    "Role": {
        "Path": "/",
        "RoleName": "student-record-system-ecs-task-execution-role",
        "RoleId": "AROATFBMPMXFV3WFLOXH5",
        "Arn": "arn:aws:iam::216989132235:role/student-record-system-ecs-task-execution-role",
        "CreateDate": "2025-07-05T04:42:42+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ecs-tasks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ],
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "student-record-system-ecs-task-execution-role"
                },
                {
                    "Key": "Environment",
                    "Value": "production"
                }
            ]
        }
    }
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Attach AWS managed policy
Ln 1, Col 1  Spaces:4  UTF-8  LF  Dockerfile  ⌂  ⌂
```

Attach AWS managed policy

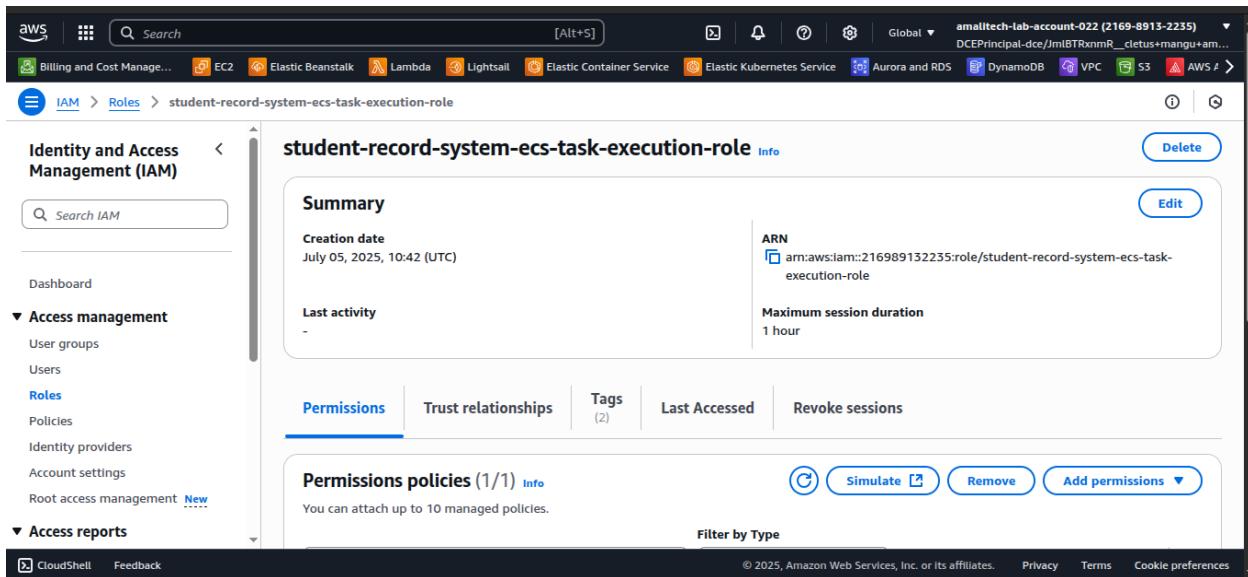
```
aws iam attach-role-policy \
    --role-name "$PROJECT_NAME-ecs-task-execution-role" \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

```
export
```

```
ECS_EXECUTION_ROLE_ARN="arn:aws:iam::$AWS_ACCOUNT_ID:role/$PROJECT_NAME-ecs-task-execution-role"
```



```
"Role": {
    "Path": "/",
    "RoleName": "student-record-system-ecs-task-execution-role",
    "RoleId": "AROATFBMPMXFV3WFLOXH5",
    "Arn": "arn:aws:iam::216989132235:role/student-record-system-ecs-task-execution-role",
    "CreateDate": "2025-07-05T10:42:42+00:00",
    "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": "ecs-tasks.amazonaws.com"
                },
                "Action": "sts:AssumeRole"
            }
        ]
    },
    "Tags": [
        {
            "Key": "Name",
            "Value": "student-record-system-ecs-task-execution-role"
        },
        {
            "Key": "Environment",
            "Value": "production"
        }
    ]
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Attach AWS managed policy
aws iam attach-role-policy \
--role-name "$PROJECT_NAME-ecs-task-execution-role" \
--policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
export ECS_EXECUTION_ROLE_ARN="arn:aws:iam::$AWS_ACCOUNT_ID:role/$PROJECT_NAME-ecs-task-execution-role"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```



The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with 'Identity and Access Management (IAM)' selected. The main area displays the 'student-record-system-ecs-task-execution-role' role. The 'Summary' section includes the ARN (arn:aws:iam::216989132235:role/student-record-system-ecs-task-execution-role), creation date (July 05, 2025, 10:42 (UTC)), and last activity (none). Below the summary, there are tabs for 'Permissions', 'Trust relationships', 'Tags (2)', 'Last Accessed', and 'Revoke sessions'. The 'Permissions' tab is active, showing one policy attached: 'Permissions policies (1/1)'. A 'Simulate' button is available for this policy.

Step 5.2: ECS Task Role

```
# Create ECS Task Role
```

```
aws iam create-role \
--role-name "$PROJECT_NAME-ecs-task-role" \
--assume-role-policy-document file://ecs-task-trust-policy.json \
--tags Key=Name,Value="$PROJECT_NAME-ecs-task-role"
```

```
Key=Environment,Value="$ENVIRONMENT"
```

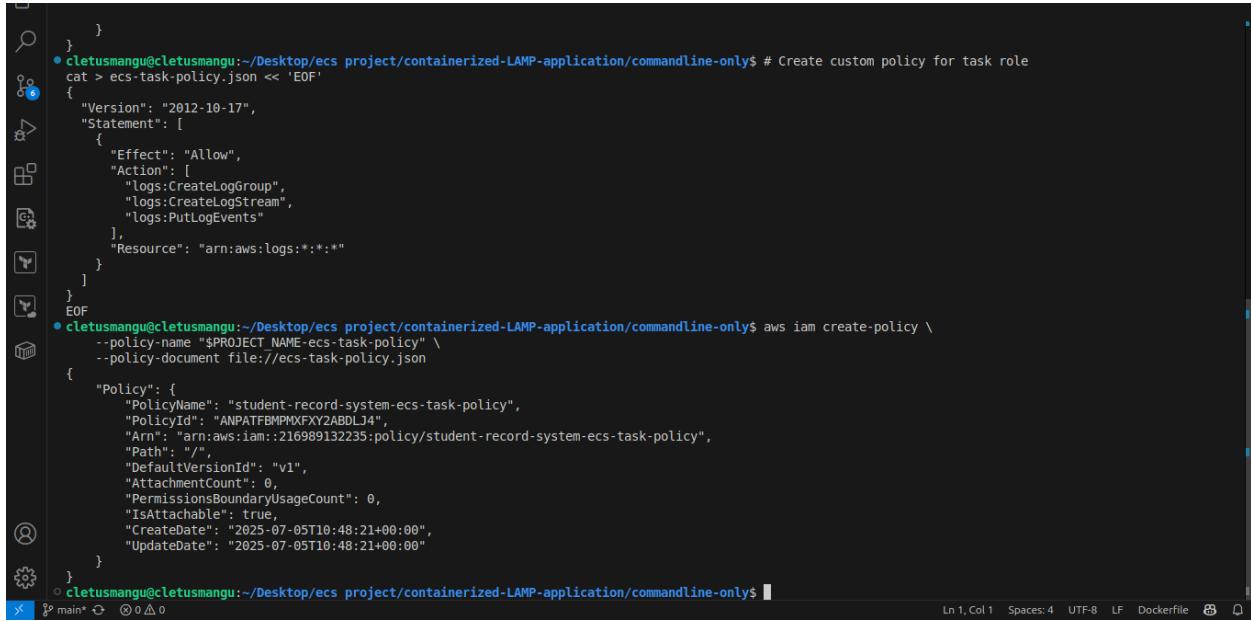
```
# Create custom policy for task role
cat > ecs-task-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    }
  ]
}
EOF
```

The screenshot shows a terminal window with the following content:

```
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws iam create-role \
  --role-name "$PROJECT_NAME-ecs-task-role" \
  --assume-role-policy-document file://ecs-task-trust-policy.json \
  --tags Key=Name,Value="$PROJECT_NAME-ecs-task-role" Key=Environment,Value="$ENVIRONMENT"
{
  "Role": {
    "Path": "/",
    "RoleName": "student-record-system-ecs-task-role",
    "RoleId": "AROATFBMMPXUFVYKKB7A",
    "Arn": "arn:aws:iam::216989132235:role/student-record-system-ecs-task-role",
    "CreateDate": "2025-07-05T0:47:14+00:00",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "ecs-tasks.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    },
    "Tags": [
      {
        "Key": "Name",
        "Value": "student-record-system-ecs-task-role"
      },
      {
        "Key": "Environment",
        "Value": "production"
      }
    ]
  }
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws iam create-policy \
  --policy-name "$PROJECT_NAME-ecs-task-policy" \
  --policy-document file://ecs-task-policy.json

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws iam attach-role-policy \
  --role-name "$PROJECT_NAME-ecs-task-role" \
```

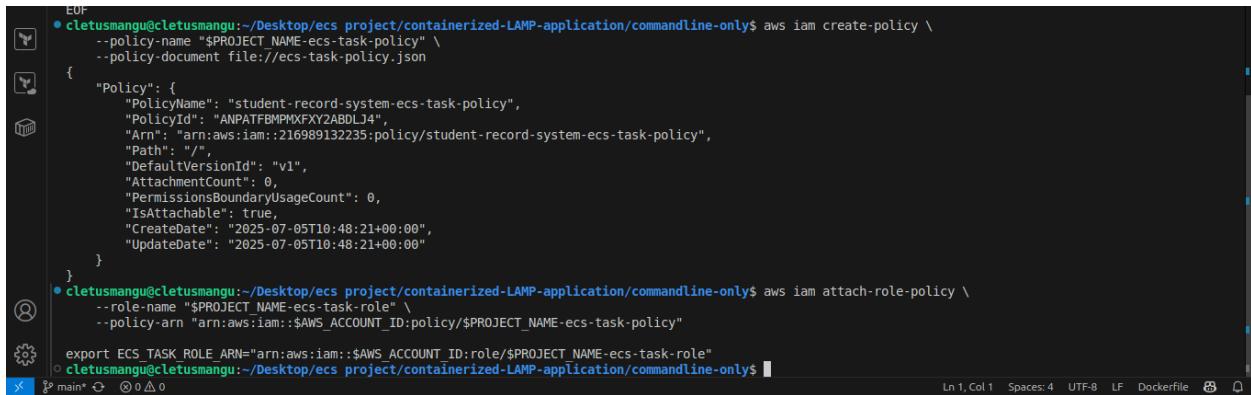
```
--policy-arn "arn:aws:iam::$AWS_ACCOUNT_ID:policy/$PROJECT_NAME-ecs-task-policy"
```



```
        }
    }
}
EOF
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create custom policy for task role
cat > ecs-task-policy.json << 'EOF'
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:/*:*/*"
        }
    ]
}
EOF
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws iam create-policy \
--policy-name "$PROJECT_NAME-ecs-task-policy" \
--policy-document file://ecs-task-policy.json
{
    "Policy": {
        "PolicyName": "student-record-system-ecs-task-policy",
        "PolicyId": "ANPATFBMPMXFY2ABDLJ4",
        "Arn": "arn:aws:iam::216989132235:policy/student-record-system-ecs-task-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2025-07-05T10:48:21+00:00",
        "UpdateDate": "2025-07-05T10:48:21+00:00"
    }
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

export

```
ECS_TASK_ROLE_ARN="arn:aws:iam::$AWS_ACCOUNT_ID:role/$PROJECT_NAME-ecs-task-role"
```



```
EOF
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws iam create-policy \
--policy-name "$PROJECT_NAME-ecs-task-policy" \
--policy-document file://ecs-task-policy.json
{
    "Policy": {
        "PolicyName": "student-record-system-ecs-task-policy",
        "PolicyId": "ANPATFBMPMXFY2ABDLJ4",
        "Arn": "arn:aws:iam::216989132235:policy/student-record-system-ecs-task-policy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2025-07-05T10:48:21+00:00",
        "UpdateDate": "2025-07-05T10:48:21+00:00"
    }
}
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws iam attach-role-policy \
--role-name "$PROJECT_NAME-ecs-task-role" \
--policy-arn "arn:aws:iam::$AWS_ACCOUNT_ID:policy/$PROJECT_NAME-ecs-task-policy"
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

Phase 6: Create Application Load Balancer

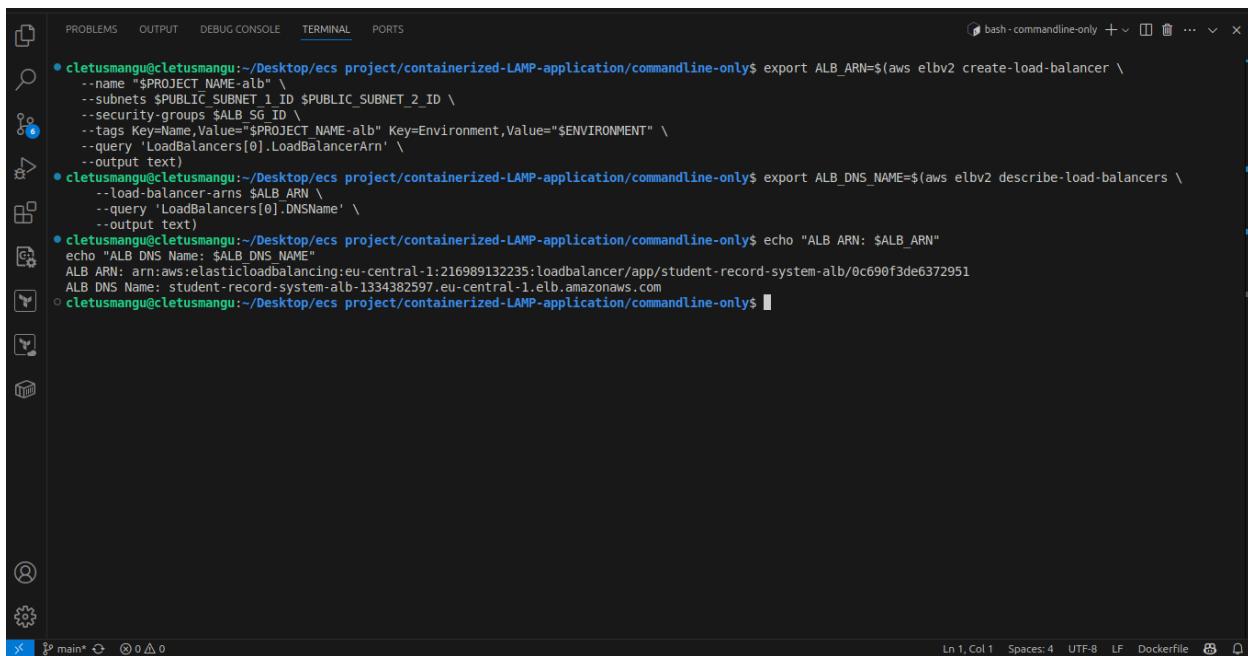
Step 6.1: Create Application Load Balancer

```
export ALB_ARN=$(aws elbv2 create-load-balancer \
--name "$PROJECT_NAME-alb" \
--subnets $PUBLIC_SUBNET_1_ID $PUBLIC_SUBNET_2_ID \
--security-groups $ALB_SG_ID \
```

```
--tags Key=Name,Value="$PROJECT_NAME-alb"
Key=Environment,Value="$ENVIRONMENT" \
--query 'LoadBalancers[0].LoadBalancerArn' \
--output text)

export ALB_DNS_NAME=$(aws elbv2 describe-load-balancers \
--load-balancer-arns $ALB_ARN \
--query 'LoadBalancers[0].DNSName' \
--output text)

echo "ALB ARN: $ALB_ARN"
echo "ALB DNS Name: $ALB_DNS_NAME"
```



The screenshot shows a terminal session in VS Code with the following history:

- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ export ALB_ARN=$(aws elbv2 create-load-balancer \
--name "$PROJECT_NAME-alb" \
--subnets $PUBLIC_SUBNET_1_ID $PUBLIC_SUBNET_2_ID \
--security-groups $ALB_SG_ID \
--tags Key=Name,Value="$PROJECT_NAME-alb" Key=Environment,Value="$ENVIRONMENT" \
--query 'LoadBalancers[0].LoadBalancerArn' \
--output text)`
- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ export ALB_DNS_NAME=$(aws elbv2 describe-load-balancers \
--load-balancer-arns $ALB_ARN \
--query 'LoadBalancers[0].DNSName' \
--output text)`
- `cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "ALB ARN: $ALB_ARN"
ALB ARN: arn:aws:elasticloadbalancing:eu-central-1:216989132235:loadbalancer/app/student-record-system-alb/0c690f3de6372951
ALB DNS Name: student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$`

The terminal interface includes standard VS Code icons for problems, output, debug console, and ports. The bottom status bar shows the file path as `main* · 0 · 0 · 0` and the current file as `Dockerfile`.

The screenshot shows the AWS CloudFront console with the following details:

- Region:** Europe (Frankfurt)
- Load Balancer:** student-record-system-alb
- Details:**
 - Load balancer type:** Application
 - Status:** Active
 - VPC:** vpc-036824e0a8e209300
 - Hosted zone:** Z215JYRZR1TBDS
 - Availability Zones:**
 - subnet-04fb597c03872f64 (eu-central-1b (euc1-az3))
 - subnet-0cffb475ccc65f5a1 (eu-central-1a (euc1-az2))
 - Load balancer ARN:** arn:aws:elasticloadbalancing:eu-central-1:216989132235:loadbalancer/app/student-record-system-alb/0c690f3de6372951
 - DNS name:** student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com (A Record)
- Listeners and rules:** Selected tab.
- Network mapping:**
- Resource map:**
- Security:**
- Monitoring:**
- Integrations:**
- Attributes:**

Step 6.2: Create Target Group

```

export TARGET_GROUP_ARN=$(aws elbv2 create-target-group \
--name "$PROJECT_NAME-tg" \
--protocol HTTP \
--port 80 \
--vpc-id $VPC_ID \
--target-type ip \
--health-check-enabled \
--health-check-interval-seconds 30 \
--health-check-path "/" \
--health-check-protocol HTTP \
--health-check-timeout-seconds 5 \
--healthy-threshold-count 2 \
--unhealthy-threshold-count 2 \
--tags Key=Name,Value="$PROJECT_NAME-tg"
Key=Environment,Value="$ENVIRONMENT" \
--query 'TargetGroups[0].TargetGroupArn' \
--output text)

echo "Target Group ARN: $TARGET_GROUP_ARN"

```

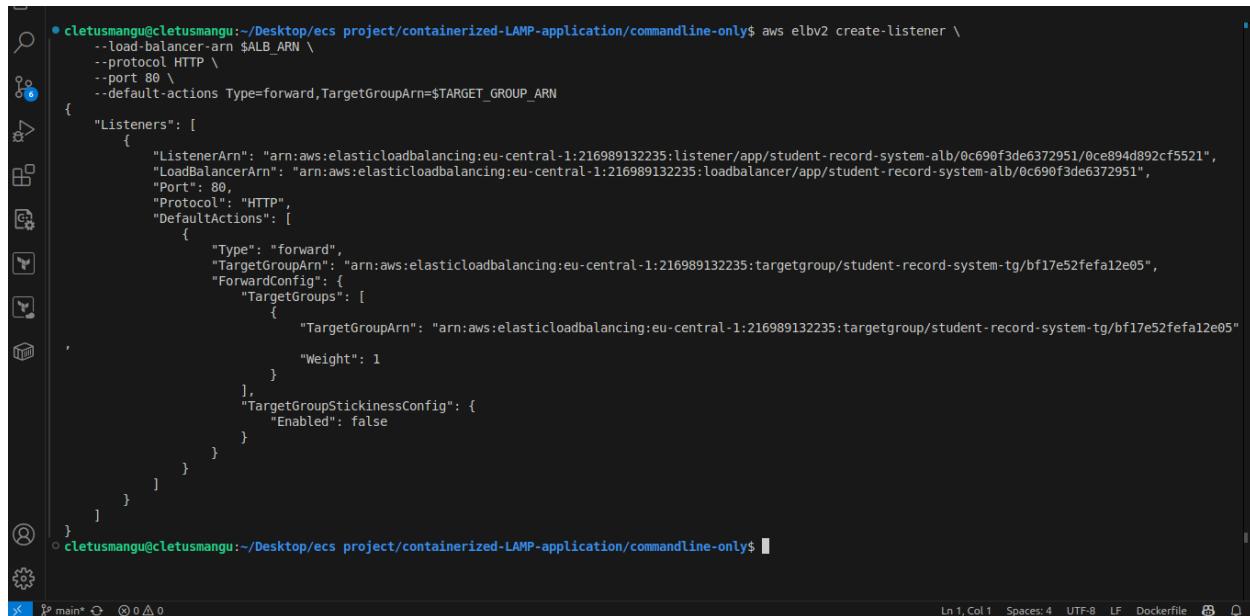
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS bash - commandline-only + ~ ⏺ ... ✘ × cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ export TARGET_GROUP_ARN=$(aws elbv2 create-target-group \
--name "$PROJECT_NAME-tg" \
--protocol HTTP \
--port 80 \
--vpc-id $VPC_ID \
--target-type ip \
--health-check-enabled \
--health-check-interval-seconds 30 \
--health-check-path "/" \
--health-check-protocol HTTP \
--health-check-timeout-seconds 5 \
--healthy-threshold-count 2 \
--unhealthy-threshold-count 2 \
--tags Key=Name,Value="$PROJECT_NAME-tg" Key=Environment,Value="$ENVIRONMENT" \
--query 'TargetGroups[0].TargetGroupArn' \
--output text)
echo "Target Group ARN: $TARGET_GROUP_ARN"
Target Group ARN: arn:aws:elasticloadbalancing:eu-central-1:216989132235:targetgroup/student-record-system-tg/bf1e52fefaf12e05
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

The screenshot shows the AWS Lambda console interface. On the left, a sidebar menu includes 'Billing and Cost Management', 'EC2' (selected), 'Elastic Beanstalk', 'Lambda', 'Lightsail', 'Elastic Container Service', 'Elastic Kubernetes Service', 'Aurora and RDS', 'DynamoDB', 'VPC', 'S3', 'AWS Lambda', and 'AWS Lambda VPC'. The main area displays the configuration for a Lambda function named 'student-record-system-tg'. The 'Handler' field is set to 'lambda.lambda_handler'. The 'Runtime' is 'Python 3.9'. Under 'Memory', 'Timeout', and 'Tracing', the values are 512 MB, 300 seconds, and 'None' respectively. In the 'Code' section, 'Edit' is selected, and the file 'index.py' is shown with the following code:

```
def lambda_handler(event, context):  
    # Add your logic here  
    return {"statusCode": 200, "body": "Hello from Lambda!"}
```

Step 6.3: Create Listener

```
aws elbv2 create-listener \
--load-balancer-arn $ALB_ARN \
--protocol HTTP \
--port 80 \
--default-actions Type=forward,TargetGroupArn=$TARGET_GROUP_ARN
```



```
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws elbv2 create-listener \
--load-balancer-arn $ALB_ARN \
--protocol HTTP \
--port 80 \
--default-actions Type=forward,TargetGroupArn=$TARGET_GROUP_ARN

{
  "Listeners": [
    {
      "ListenerArn": "arn:aws:elasticloadbalancing:eu-central-1:216989132235:listener/app/student-record-system-alb/0c690f3de6372951/0ce894d892cf5521",
      "LoadBalancerArn": "arn:aws:elasticloadbalancing:eu-central-1:216989132235:loadbalancer/app/student-record-system-alb/0c690f3de6372951",
      "Port": 80,
      "Protocol": "HTTP",
      "DefaultActions": [
        {
          "Type": "forward",
          "TargetGroupArn": "arn:aws:elasticloadbalancing:eu-central-1:216989132235:targetgroup/student-record-system-tg/bf17e52fefaf12e05",
          "ForwardConfig": {
            "TargetGroups": [
              {
                "TargetGroupArn": "arn:aws:elasticloadbalancing:eu-central-1:216989132235:targetgroup/student-record-system-tg/bf17e52fefaf12e05"
                "Weight": 1
              }
            ],
            "TargetGroupStickinessConfig": {
              "Enabled": false
            }
          }
        }
      ]
    }
  ]
}

cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

Phase 7: Create ECS Infrastructure

Step 7.1: Create CloudWatch Log Group

```
aws logs create-log-group \
--log-group-name "/aws/ecs/$PROJECT_NAME" \
--retention-in-days 7 \
--tags Key=Name,Value="$PROJECT_NAME-log-group"
Key=Environment,Value="$ENVIRONMENT"
```

```

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws logs create-log-group \
--log-group-name "/aws/ecs/$PROJECT_NAME"

An error occurred (ResourceAlreadyExistsException) when calling the CreateLogGroup operation: The specified log group already exists
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws logs put-retention-policy \
--log-group-name "/aws/ecs/$PROJECT_NAME" \
--retention-in-days 7
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws logs tag-log-group \
--log-group-name "/aws/ecs/$PROJECT_NAME" \
--tags Key=Name,Value="$PROJECT_NAME-log-group" Key=Environment,Value="$ENVIRONMENT"

usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:
aws help
aws <command> help
aws <command> <subcommand> help

Unknown options: Key=Environment,Value=production

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws logs tag-log-group \
--log-group-name "/aws/ecs/$PROJECT_NAME" \
--tags '[{"Name": "'"$PROJECT_NAME"'-log-group", "Environment": "'"$ENVIRONMENT"'"}]'
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws logs describe-log-groups --log-group-name-prefix "/aws/ecs/$PROJECT_NAME"
{
    "logGroups": [
        {
            "logGroupName": "/aws/ecs/student-record-system",
            "creationTime": 1751713621805,
            "retentionInDays": 7,
            "metricFilterCount": 0,
            "arn": "arn:aws:logs:eu-central-1:216989132235:log-group:/aws/ecs/student-record-system:*",
            "storedBytes": 0,
            "logGroupClass": "STANDARD",
            "logGroupArn": "arn:aws:logs:eu-central-1:216989132235:log-group:/aws/ecs/student-record-system"
        }
    ]
}

```

Ln 1, Col 1 Spaces: 4 UTF-8 LF Dockerfile 8 Q

The screenshot shows the AWS CloudWatch Log Groups interface. The left sidebar has sections for Alarms, Logs (selected), Metrics, Application Signals (APM), and Network Monitoring. Under Logs, there are sub-options for Log Anomalies, Live Tail, Logs Insights, and Contributor Insights. The main content area shows the log group details for '/aws/ecs/student-record-system'. The details pane includes sections for Log class (Standard), Metric filters (0), Data protection (Sensitive data count -), ARN (arn:aws:logs:eu-central-1:216989132235:log-group:/aws/ecs/student-record-system-*), Subscription filters (0), Contributor Insights rules (-), Field indexes (Configure), Creation time (3 minutes ago), KMS key ID (-), Transformer (Configure), Retention (1 week), Anomaly detection (Configure), and Stored bytes (-). At the bottom, there are links for CloudShell and Feedback, and a footer with copyright information.

Step 7.2: Create ECS Cluster

```

export ECS_CLUSTER_ARN=$(aws ecs create-cluster \
--cluster-name "$PROJECT_NAME-cluster" \
--capacity-providers FARGATE \
--default-capacity-provider-strategy capacityProvider=FARGATE,weight=1 \
--configuration
executeCommandConfiguration='{logging=OVERRIDE,logConfiguration={cloudWatchLogGro \
upName=/aws/ecs/'$PROJECT_NAME'}}' \
--tags key=Name,value="$PROJECT_NAME-cluster"

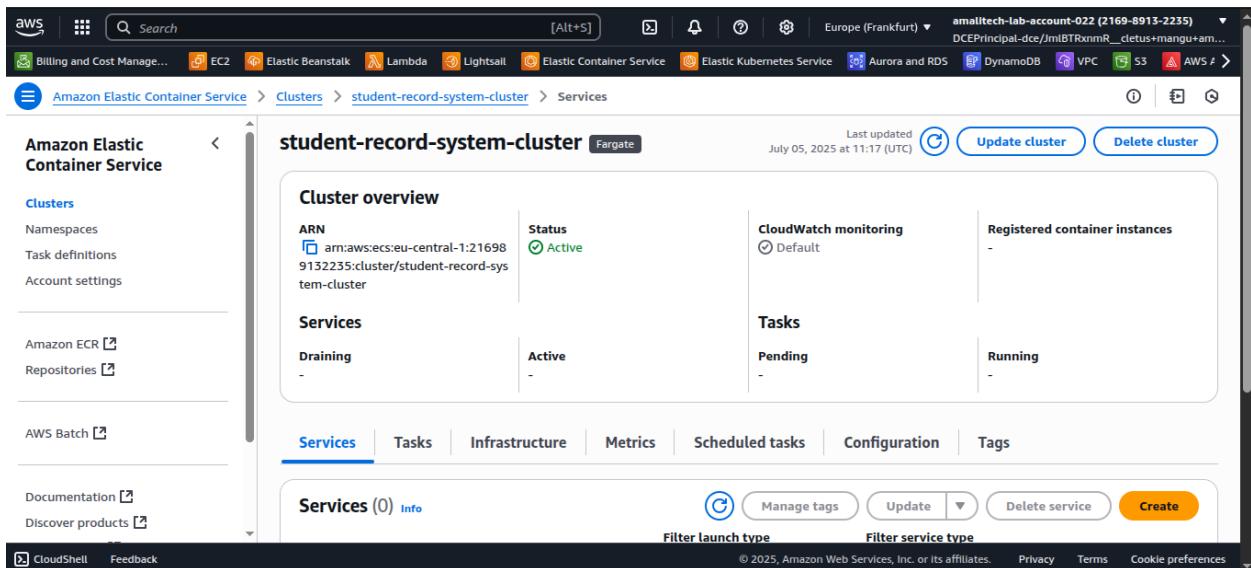
```

```
key=Environment,value="$ENVIRONMENT" \
--query 'cluster.clusterArn' \
--output text)
```

```
echo "ECS Cluster ARN: $ECS_CLUSTER_ARN"
```

```
• cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ export ECS_CLUSTER_ARN=$(aws ecs create-cluster \
--cluster-name "$PROJECT_NAME-cluster" \
--capacity-providers FARGATE \
--default-capacity-provider-strategy capacityProvider=FARGATE,weight=1 \
--configuration '{executeCommandConfiguration: {"logging": "OVERRIDE", "logConfiguration": {"cloudWatchLogGroupName": "/aws/ecs/'$PROJECT_NAME'"}}}' \
--tags '[{"key": "Name", "value": "'"$PROJECT_NAME-cluster"'"}, {"key": "Environment", "value": "'"$ENVIRONMENT"'"}]' \
--query 'cluster.clusterArn' \
--output text)

echo "ECS Cluster ARN: $ECS_CLUSTER_ARN"
ECS Cluster ARN: arn:aws:ec2:eu-central-1:216989132235:cluster/student-record-system-cluster
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```



Step 7.3: Create Task Definition

```
cat > task-definition.json << EOF
{
  "family": "$PROJECT_NAME",
  "networkMode": "awsvpc",
  "requiresCompatibilities": ["FARGATE"],
  "cpu": "256",
  "memory": "512",
  "executionRoleArn": "$ECS_EXECUTION_ROLE_ARN",
  "taskRoleArn": "$ECS_TASK_ROLE_ARN",
  "containerDefinitions": [
```

```
{  
  "name": "$PROJECT_NAME",  
  "image": "$ECR_REPOSITORY_URI:latest",  
  "portMappings": [  
    {  
      "containerPort": 80,  
      "protocol": "tcp"  
    }  
  ],  
  "environment": [  
    {  
      "name": "DB_HOST",  
      "value": "$DB_ENDPOINT"  
    },  
    {  
      "name": "DB_USER",  
      "value": "$DB_USERNAME"  
    },  
    {  
      "name": "DB_PASSWORD",  
      "value": "$DB_PASSWORD"  
    },  
    {  
      "name": "DB_NAME",  
      "value": "$DB_NAME"  
    },  
    {  
      "name": "DB_PORT",  
      "value": "3306"  
    }  
  ],  
  "logConfiguration": {  
    "logDriver": "awslogs",  
    "options": {  
      "awslogs-group": "/aws/ecs/$PROJECT_NAME",  
      "awslogs-region": "$AWS_REGION",  
      "awslogs-stream-prefix": "ecs"  
    }  
  },  
  "essential": true
```

```
}
```

```
]
```

```
}
```

```
EOF
```

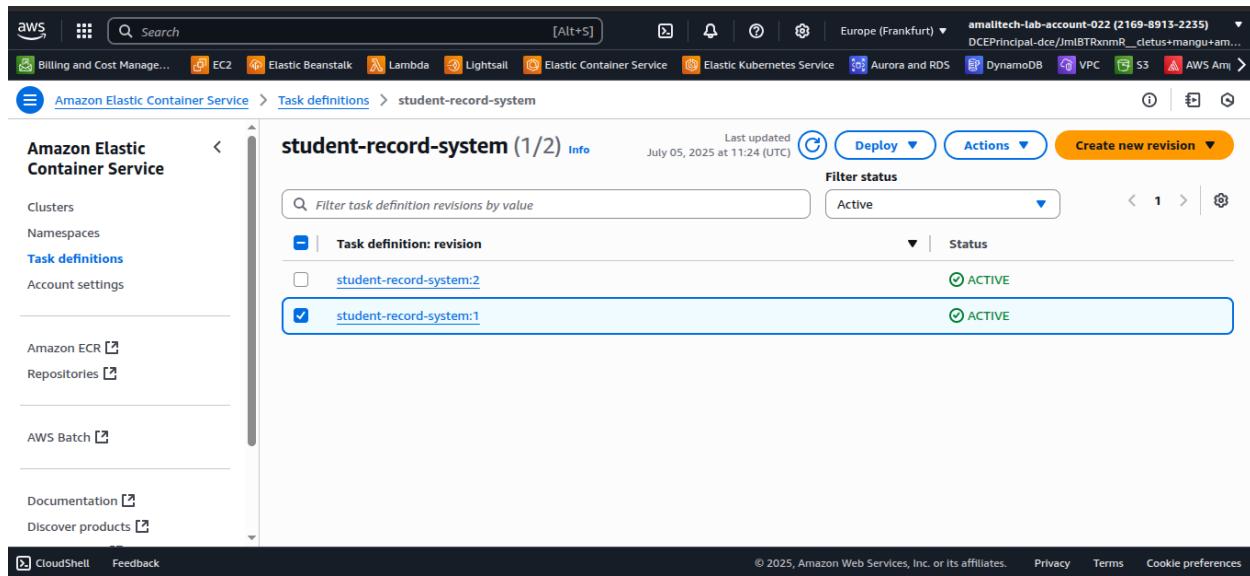
The screenshot shows a terminal window with the following content:

```
cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ cat > task-definition.json <<EOF
{
  "family": "$PROJECT_NAME",
  "networkMode": "awsvpc",
  "requiresCompatibilities": ["FARGATE"],
  "cpu": "256",
  "memory": "512",
  "executionRoleArn": "$ECS_EXECUTION_ROLE_ARN",
  "taskRoleArn": "$ECS_TASK_ROLE_ARN",
  "containerDefinitions": [
    {
      "name": "$PROJECT_NAME",
      "image": "$ECR_REPOSITORY_URI:latest",
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "environment": [
        {
          "name": "DB_HOST",
          "value": "$DB_ENDPOINT"
        },
        {
          "name": "DB_USER",
          "value": "$DB_USERNAME"
        },
        {
          "name": "DB_PASSWORD",
          "value": "$DB_PASSWORD"
        },
        {
          "name": "DB_NAME",
          "value": "$DB_NAME"
        }
      ]
    }
  ]
}
EOF
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF Dockerfile 8 Q

```
export TASK_DEFINITION_ARN=$(aws ecs register-task-definition \
--cli-input-json file://task-definition.json \
--tags key=Name,value="$PROJECT_NAME-task-definition"
key=Environment,value="$ENVIRONMENT" \
--query 'taskDefinition.taskDefinitionArn' \
--output text)
```

```
echo "Task Definition ARN: $TASK_DEFINITION_ARN"
```



Step 7.4: Create ECS Service

```
export ECS_SERVICE_ARN=$(aws ecs create-service \
--cluster "$PROJECT_NAME-cluster" \
--service-name "$PROJECT_NAME-service" \
--task-definition "$PROJECT_NAME" \
--desired-count 2 \
--launch-type FARGATE \
--network-configuration
"awsvpcConfiguration={subnets=[${PRIVATE_SUBNET_1_ID},${PRIVATE_SUBNET_2_ID}],se
curityGroups=[${ECS_SG_ID}],assignPublicIp=DISABLED}" \
--load-balancers
targetGroupArn=$TARGET_GROUP_ARN,containerName=$PROJECT_NAME,containerPort
=80 \
--tags key=Name,value="$PROJECT_NAME-service"
key=Environment,value="$ENVIRONMENT" \
--query 'service.serviceArn' \
--output text)

echo "ECS Service ARN: $ECS_SERVICE_ARN"
```

```

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ export ECS_SERVICE_ARN=$(aws ecs create-service \
--cluster "$PROJECT_NAME-cluster" \
--service-name "$PROJECT_NAME-service" \
--task-definition "$PROJECT_NAME" \
--desired-count 2 \
--launch-type FARGATE \
--network-configuration "awsvpcConfiguration={subnets=[${PRIVATE_SUBNET_1_ID},${PRIVATE_SUBNET_2_ID}],securityGroups=[${ECS_SG_ID}],assignPublicIp=DISABLED}" \
--load-balancers targetGroupArn=${TARGET_GROUP_ARN},containerName=$PROJECT_NAME,containerPort=80 \
--tags key=Name,value="$PROJECT_NAME-service" key=Environment,value="$ENVIRONMENT" \
--query 'service.serviceArn' \
--output text)
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "ECS Service ARN: $ECS_SERVICE_ARN"
ECS Service ARN: arn:aws:ecs:eu-central-1:216989132235:service/student-record-system-cluster/student-record-system-service
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ 

```

The screenshot shows the AWS Management Console interface for the Amazon Elastic Container Service (ECS). The left sidebar navigation bar includes links for Billing and Cost Management, EC2, Elastic Beanstalk, Lambda, Lightsail, Elastic Container Service, Elastic Kubernetes Service, Aurora and RDS, DynamoDB, VPC, S3, and AWS Lambda. The main content area displays the 'student-record-system-service' details under the 'Clusters > student-record-system-cluster > Services' path. The service overview shows 2 desired tasks (1 active, 1 pending), a task definition revision, and an in-progress deployment. The status section provides details like Service name (student-record-system-service), Service ARN (arn:aws:ecs:eu-central-1:216989132235:service/student-record-system-cluster/student-record-system-service), Deployments current state (1 in progress task, 1 completed task), and Creation date (July 05, 2025 at 11:25 (UTC)).

Step 7.5: Setup Auto Scaling

```
# Register scalable target
aws application-autoscaling register-scalable-target \
--service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id "service/$PROJECT_NAME-cluster/$PROJECT_NAME-service" \
--min-capacity 1 \
--max-capacity 10
```

```
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Register scalable target
aws application-autoscaling register-scalable-target \
--service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id "service/$PROJECT_NAME-cluster/$PROJECT_NAME-service" \
--min-capacity 1 \
--max-capacity 10
{
  "scalableTargetARN": "arn:aws:application-autoscaling:eu-central-1:216989132235:scalable-target/0ec50accb3750534606ad5661b36a02702e"
}
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create scaling policy
aws application-autoscaling put-scaling-policy \
--service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id "service/$PROJECT_NAME-cluster/$PROJECT_NAME-service" \
--policy-name "$PROJECT_NAME-cpu-scaling" \
--policy-type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration '{
    "TargetValue": 70.0,
    "PredefinedMetricSpecification": {
        "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
    }
}'
{
  "PolicyARN": "arn:aws:autoscaling:eu-central-1:216989132235:scalingPolicy:0caccb37-5053-4606-ad56-61b36a02702e:resource/ecs/service/student-record-system-cluster/student-record-system-service:policyName/student-record-system-cpu-scaling",
  "Alarms": [
    {
      "AlarmName": "TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmHigh-23e7a277-84b3-46cb-a00d-719c222cd7e4",
      "AlarmARN": "arn:aws:cloudwatch:eu-central-1:216989132235:alarm:TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmHigh-23e7a277-84b3-46cb-a00d-719c222cd7e4"
    },
    {
      "AlarmName": "TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmLow-e6a16772-87da-4a37-84c6-5882e656374f",
      "AlarmARN": "arn:aws:cloudwatch:eu-central-1:216989132235:alarm:TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmLow-e6a16772-87da-4a37-84c6-5882e656374f"
    }
  ]
}
Ln 1, Col 1  Spaces: 4  UTF-8  LF  Dockerfile  ⚙  📁
```

Create scaling policy

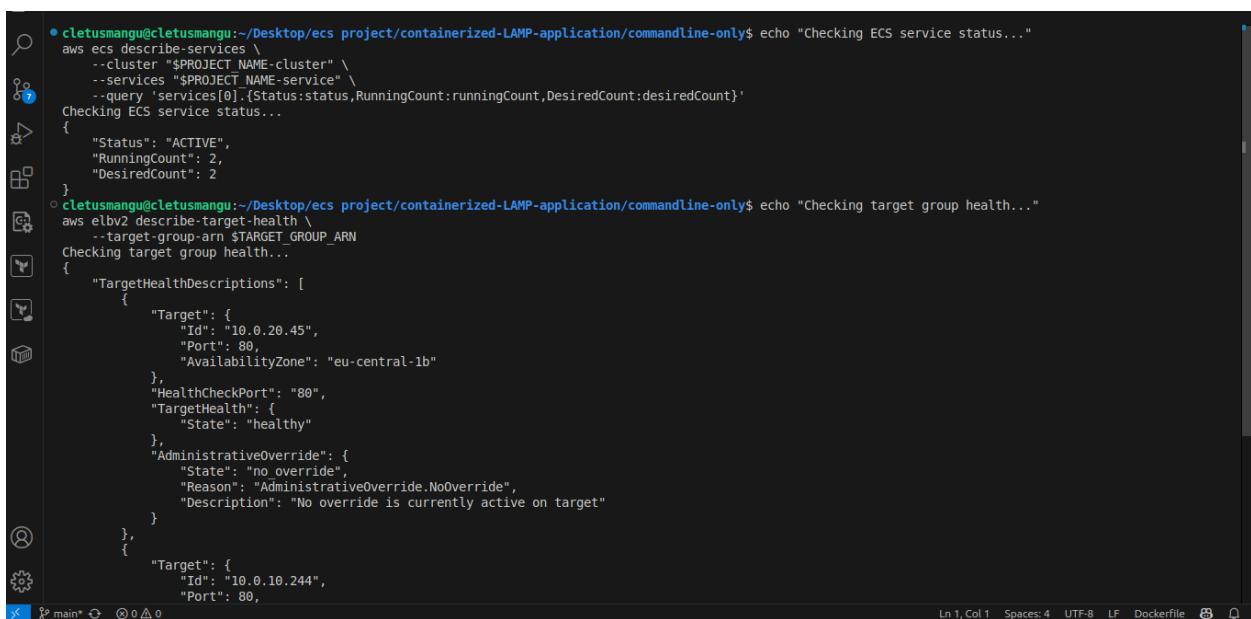
```
aws application-autoscaling put-scaling-policy \
--service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id "service/$PROJECT_NAME-cluster/$PROJECT_NAME-service" \
--policy-name "$PROJECT_NAME-cpu-scaling" \
--policy-type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration '{
    "TargetValue": 70.0,
    "PredefinedMetricSpecification": {
        "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
    }
}'
```

```
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # Create scaling policy
aws application-autoscaling put-scaling-policy \
--service-namespace ecs \
--scalable-dimension ecs:service:DesiredCount \
--resource-id "service/$PROJECT_NAME-cluster/$PROJECT_NAME-service" \
--policy-name "$PROJECT_NAME-cpu-scaling" \
--policy-type TargetTrackingScaling \
--target-tracking-scaling-policy-configuration '{
    "TargetValue": 70.0,
    "PredefinedMetricSpecification": {
        "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
    }
}'
{
  "PolicyARN": "arn:aws:autoscaling:eu-central-1:216989132235:scalingPolicy:0caccb37-5053-4606-ad56-61b36a02702e:resource/ecs/service/student-record-system-cluster/student-record-system-service:policyName/student-record-system-cpu-scaling",
  "Alarms": [
    {
      "AlarmName": "TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmHigh-23e7a277-84b3-46cb-a00d-719c222cd7e4",
      "AlarmARN": "arn:aws:cloudwatch:eu-central-1:216989132235:alarm:TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmHigh-23e7a277-84b3-46cb-a00d-719c222cd7e4"
    },
    {
      "AlarmName": "TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmLow-e6a16772-87da-4a37-84c6-5882e656374f",
      "AlarmARN": "arn:aws:cloudwatch:eu-central-1:216989132235:alarm:TargetTracking-service/student-record-system-cluster/student-record-system-service-AlarmLow-e6a16772-87da-4a37-84c6-5882e656374f"
    }
  ]
}
● cletusmangu@cletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

Phase 8: Verify Deployment

Step 8.1: Check Service Status

```
echo "Checking ECS service status..."  
aws ecs describe-services \  
  --cluster "$PROJECT_NAME-cluster" \  
  --services "$PROJECT_NAME-service" \  
  --query 'services[0].{Status:status,RunningCount:runningCount,DesiredCount:desiredCount}'  
  
echo "Checking target group health..."  
aws elbv2 describe-target-health \  
  --target-group-arn $TARGET_GROUP_ARN  
  
echo "Application will be available at: http://\$ALB\_DNS\_NAME"
```



The screenshot shows a terminal window with the following content:

```
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Checking ECS service status..."  
aws ecs describe-services \  
  --cluster "$PROJECT_NAME-cluster" \  
  --services "$PROJECT_NAME-service" \  
  --query 'services[0].{Status:status,RunningCount:runningCount,DesiredCount:desiredCount}'  
Checking ECS service status...  
{  
  "Status": "ACTIVE",  
  "RunningCount": 2,  
  "DesiredCount": 2  
}  
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Checking target group health..."  
aws elbv2 describe-target-health \  
  --target-group-arn $TARGET_GROUP_ARN  
Checking target group health...  
{  
  "TargetHealthDescriptions": [  
    {  
      "Target": {  
        "Id": "10.0.20.45",  
        "Port": 80,  
        "AvailabilityZone": "eu-central-1b"  
      },  
      "HealthCheckPort": "80",  
      "TargetHealth": {  
        "State": "healthy"  
      },  
      "AdministrativeOverride": {  
        "State": "no override",  
        "Reason": "AdministrativeOverride.NoOverride",  
        "Description": "No override is currently active on target"  
      }  
    },  
    {  
      "Target": {  
        "Id": "10.0.10.244",  
        "Port": 80,  
        "AvailabilityZone": "eu-central-1a"  
      },  
      "HealthCheckPort": "80",  
      "TargetHealth": {  
        "State": "unhealthy",  
        "Reason": "AdministrativeOverride.NoOverride",  
        "Description": "No override is currently active on target"  
      }  
    }  
  ]  
}
```

The terminal window has a dark background with light-colored text. It includes standard terminal icons on the left and a status bar at the bottom with information like 'Ln 1, Col 1' and file names.

```

cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Checking target group health..."
aws elbv2 describe-target-health \
  --target-group-arn $TARGET_GROUP_ARN
Checking target group health...
{
  "TargetHealthDescriptions": [
    {
      "Target": {
        "Id": "10.0.20.45",
        "Port": 80,
        "AvailabilityZone": "eu-central-1b"
      },
      "HealthCheckPort": "80",
      "TargetHealth": {
        "State": "healthy"
      },
      "AdministrativeOverride": {
        "State": "no override",
        "Reason": "AdministrativeOverride.NoOverride",
        "Description": "No override is currently active on target"
      }
    },
    {
      "Target": {
        "Id": "10.0.10.244",
        "Port": 80,
        "AvailabilityZone": "eu-central-1a"
      },
      "HealthCheckPort": "80",
      "TargetHealth": {
        "State": "healthy"
      },
      "AdministrativeOverride": {
        "State": "no override",
        "Reason": "AdministrativeOverride.NoOverride",
        "Description": "No override is currently active on target"
      }
    }
  ]
}
cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Application will be available at: http://$ALB_DNS_NAME"
Application will be available at: http://student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com
cletusmangu@CletusMangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ 

```

Step 8.2: Wait for Deployment

```
echo "Waiting for service to become stable..."
```

```
aws ecs wait services-stable \
--cluster "$PROJECT_NAME-cluster" \
--services "$PROJECT_NAME-service"
```

```
echo "Deployment completed successfully!"
echo "Application URL: http://\$ALB\_DNS\_NAME"
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash - Commandline-only + v ⌂ ⌂ ... v x

• cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Waiting for service to become stable..."
aws ecs wait services-stable \
--cluster "$PROJECT_NAME-cluster" \
--services "$PROJECT_NAME-service"
Waiting for service to become stable...
• cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Waiting for service to become stable..."
aws ecs wait services-stable \
--cluster "$PROJECT_NAME-cluster" \
--services "$PROJECT_NAME-service"
Waiting for service to become stable...
• cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Waiting for service to become stable..."
aws ecs wait services-stable \
--cluster "$PROJECT_NAME-cluster" \
--services "$PROJECT_NAME-service"

echo "Deployment completed successfully!"
echo "Application URL: http://$LB_DNS_NAME"
Waiting for service to become stable...
Deployment completed successfully!
Application URL: http://student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ 

```

Not secure student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com/index.php

Add New Student

ID	Name	Age	Department	Added On	Actions
1	Cletus	40	Computer Science	Jul 05, 2025	<button>Delete</button>

Phase 9: Optional - Setup Route 53 DNS

Step 9.1: Create Route 53 Record (if you have a domain)

```
# Only run this if you have a hosted zone in Route 53
# export DOMAIN_NAME="yourdomain.com"
```

```

# export SUBDOMAIN="app"
#
# export HOSTED_ZONE_ID=$(aws route53 list-hosted-zones \
#   --query "HostedZones[?Name=='${DOMAIN_NAME}'.].Id" \
#   --output text | cut -d'/' -f3)
#
# cat > dns-record.json << EOF
# {
#   "Comment": "Create A record for ALB",
#   "Changes": [
#     {
#       "Action": "CREATE",
#       "ResourceRecordSet": {
#         "Name": "$SUBDOMAIN.$DOMAIN_NAME",
#         "Type": "A",
#         "AliasTarget": {
#           "DNSName": "$ALB_DNS_NAME",
#           "EvaluateTargetHealth": true,
#           "HostedZoneId": "$(aws elbv2 describe-load-balancers --load-balancer-arns
# $ALB_ARN --query 'LoadBalancers[0].CanonicalHostedZoneId' --output text)"
#         }
#       }
#     }
#   ]
# }
# EOF
#
# aws route53 change-resource-record-sets \
#   --hosted-zone-id $HOSTED_ZONE_ID \
#   --change-batch file://dns-record.json

```

Phase 10: Setup Monitoring and Alarms

Step 10.1: Create CloudWatch Alarms

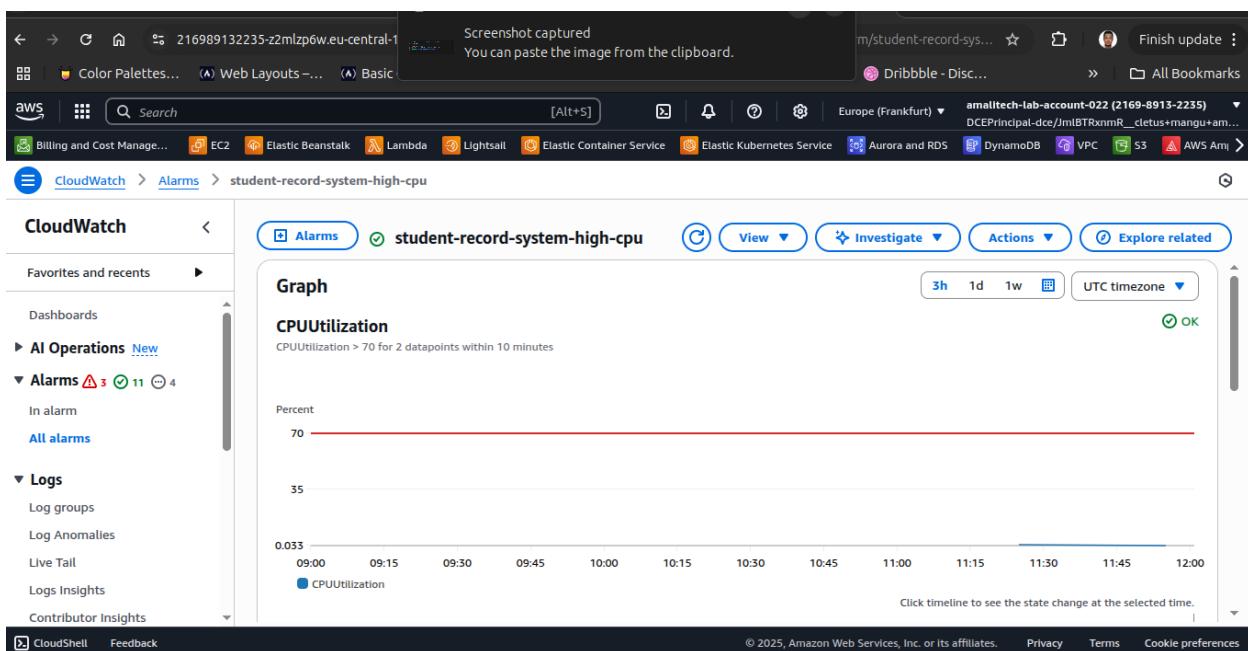
```

# CPU Utilization Alarm
aws cloudwatch put-metric-alarm \
  --alarm-name "$PROJECT_NAME-high-cpu" \

```

```
--alarm-description "High CPU utilization for ECS service" \
--metric-name CPUUtilization \
--namespace AWS/ECS \
--statistic Average \
--period 300 \
--threshold 70.0 \
--comparison-operator GreaterThanThreshold \
--evaluation-periods 2 \
--dimensions Name=ServiceName,Value="$PROJECT_NAME-service"
Name=ClusterName,Value="$PROJECT_NAME-cluster"
```

```
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws cloudwatch put-metric-alarm \
--alarm-name "$PROJECT_NAME-high-cpu" \
--alarm-description "High CPU utilization for ECS service" \
--metric-name CPUUtilization \
--namespace AWS/ECS \
--statistic Average \
--period 300 \
--threshold 70.0 \
--comparison-operator GreaterThanThreshold \
--evaluation-periods 2 \
--dimensions Name=ServiceName,Value="$PROJECT_NAME-service" Name=ClusterName,Value="$PROJECT_NAME-cluster" \
--alarm-actions arn:aws:sns:eu-central-1:123456789012:NotifyMe
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```

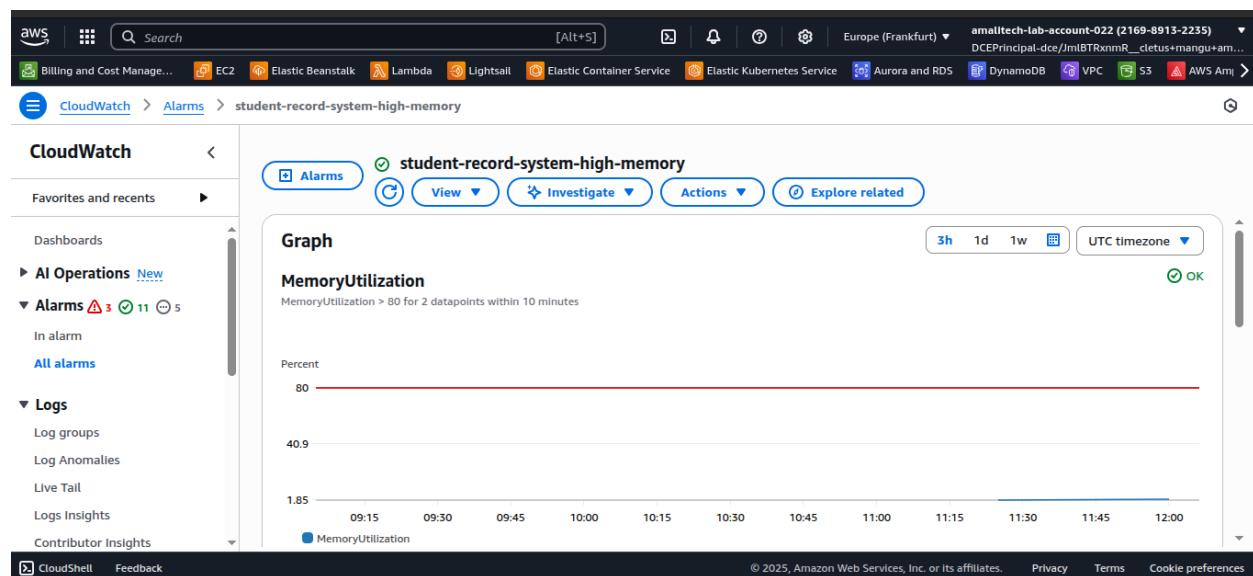


```
# Memory Utilization Alarm
aws cloudwatch put-metric-alarm \
--alarm-name "$PROJECT_NAME-high-memory" \
--alarm-description "High memory utilization for ECS service" \
--metric-name MemoryUtilization \
```

```
--namespace AWS/ECS \
--statistic Average \
--period 300 \
--threshold 80.0 \
--comparison-operator GreaterThanThreshold \
--evaluation-periods 2 \
--dimensions Name=ServiceName,Value="$PROJECT_NAME-service"
Name=ClusterName,Value="$PROJECT_NAME-cluster"
```

```
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws cloudwatch put-metric-alarm \
--alarm-name "$PROJECT_NAME-high-memory"
--alarm-description "High memory utilization for ECS service"
--metric-name MemoryUtilization
--namespace AWS/ECS
--statistic Average
--period 300
--threshold 80.0
--comparison-operator GreaterThanThreshold
--evaluation-periods 2
--dimensions Name=ServiceName,Value="$PROJECT_NAME-service" Name=ClusterName,Value="$PROJECT_NAME-cluster"
--treat-missing-data notBreaching
--alarm-actions arn:aws:sns:eu-central-1:123456789012:NotifyMe
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws sns create-topic --name NotifyMe
{
  "TopicArn": "arn:aws:sns:eu-central-1:216989132235:NotifyMe"
}
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws sns subscribe \
--topic-arn arn:aws:sns:eu-central-1:123456789012:NotifyMe \
--protocol email \
--notification-endpoint mangucleitus@gmail.com

An error occurred (InvalidClientTokenId) when calling the Subscribe operation: No account found for the given parameters
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws sns subscribe \
--topic-arn arn:aws:sns:eu-central-1:216989132235:NotifyMe \
--protocol email \
--notification-endpoint mangucleitus@gmail.com
{
  "SubscriptionArn": "pending confirmation"
}
cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$
```



AWS Notification - Subscription Confirmation ➔ Inbox



AWS Notifications Inbox

to me ▾

12:05 PM (1 minute ago)



You have chosen to subscribe to the topic:

arn:aws:sns:eu-central-1:216989132235:NotifyMe

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

Reply

Forward



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:eu-central-1:216989132235:NotifyMe:b5ad2cbf-9d55-40fa-b7b0-53b96863ae96

If it was not your intention to subscribe, [click here to unsubscribe](#).

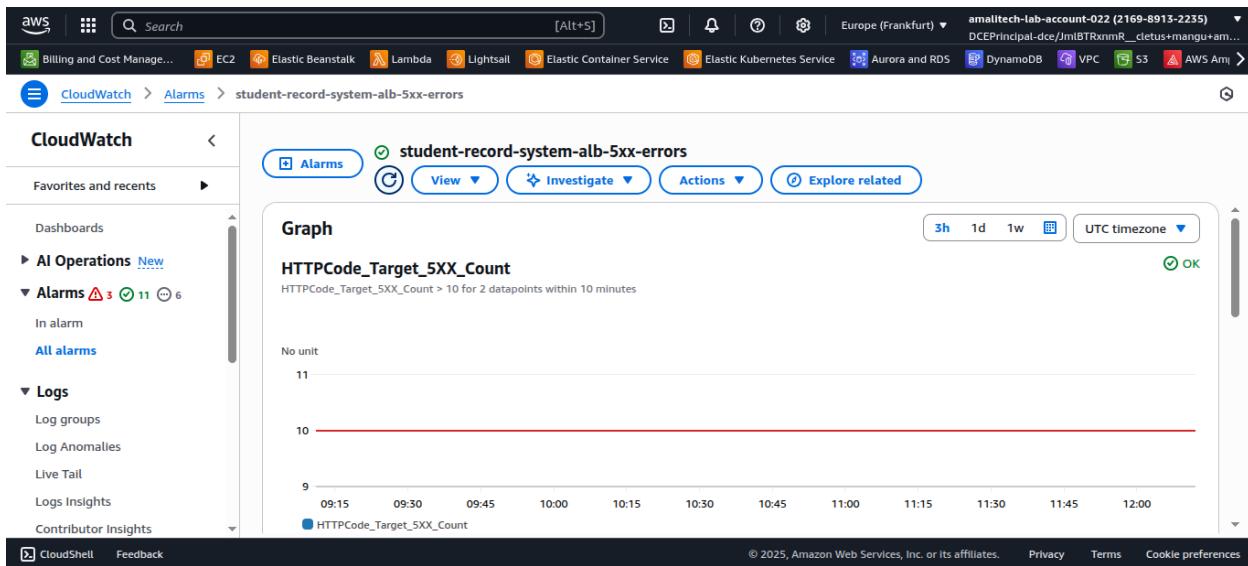
```
# ALB 5xx Error Alarm
```

```
aws cloudwatch put-metric-alarm \
--alarm-name "$PROJECT_NAME-alb-5xx-errors" \
--alarm-description "High 5xx error rate from ALB" \
--metric-name HTTPCode_Target_5XX_Count \
--namespace AWS/ApplicationELB \
--statistic Sum \
--period 300 \
--threshold 10 \
--comparison-operator GreaterThanThreshold \
--evaluation-periods 2 \
--dimensions Name=LoadBalancer,Value="$(echo $ALB_ARN | cut -d'/' -f2-)"
```

```

● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ aws cloudwatch put-metric-alarm \
--alarm-name "$PROJECT_NAME-alb-5xx-errors" \
--alarm-description "High 5xx error rate from ALB" \
--metric-name HTTPCode_Target_5XX_Count \
--namespace AWS/ApplicationELB \
--statistic Sum \
--period 300 \
--threshold 10 \
--comparison-operator GreaterThanThreshold \
--evaluation-periods 2 \
--dimensions Name=LoadBalancer,Value="$(echo $ALB_ARN | cut -d'/' -f2-)" \
--treat-missing-data notBreaching \
--alarm-actions arn:aws:sns:eu-central-1:216989132235:NotifyMe
○ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ 

```



Final Verification and Testing

Test Application

```

echo "Testing application...""
curl -I http://\$ALB\_DNS\_NAME

```

```

echo "Load testing (install apache2-utils first)..."
# ab -n 100 -c 10 http://\$ALB\_DNS\_NAME

```

```

● cletusmangu@letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Testing application..." | curl -I http://$ALB_DNS_NAME
echo "Load testing (install apache2-utils first)...
# ab -n 100 -c 10 http://$ALB_DNS_NAME/
Testing application...
HTTP/1.1 200 OK
Date: Sat, 05 Jul 2025 12:14:30 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
Server: Apache/2.4.62 (Debian)
X-Powered-By: PHP/8.1.33
Load testing (install apache2-utils first)...
● cletusmangu@letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ ab -n 100 -c 10 http://$ALB_DNS_NAME/
This is ApacheBench, Version 2.3 <Revision: 1903618 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/
Benchmarking student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com (be patient)....done
● cletusmangu@letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ ab -n 100 -c 10 http://$ALB_DNS_NAME/
Server Software:      Apache/2.4.62
Server Hostname:       student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com
Server Port:          80
Document Path:        /
Document Length:     3854 bytes
Concurrency Level:    10
Time taken for tests: 4.554 seconds
Complete requests:   100
Failed requests:    0
Total transferred:  407200 bytes
HTML transferred: 385400 bytes
Requests per second: 21.96 [#/sec] (mean)
Time per request:   455.447 [ms] (mean)
● cletusmangu@letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ 

```

View Logs

echo "Recent application logs:"
aws logs tail /aws/ecs/\$PROJECT_NAME --since 30m

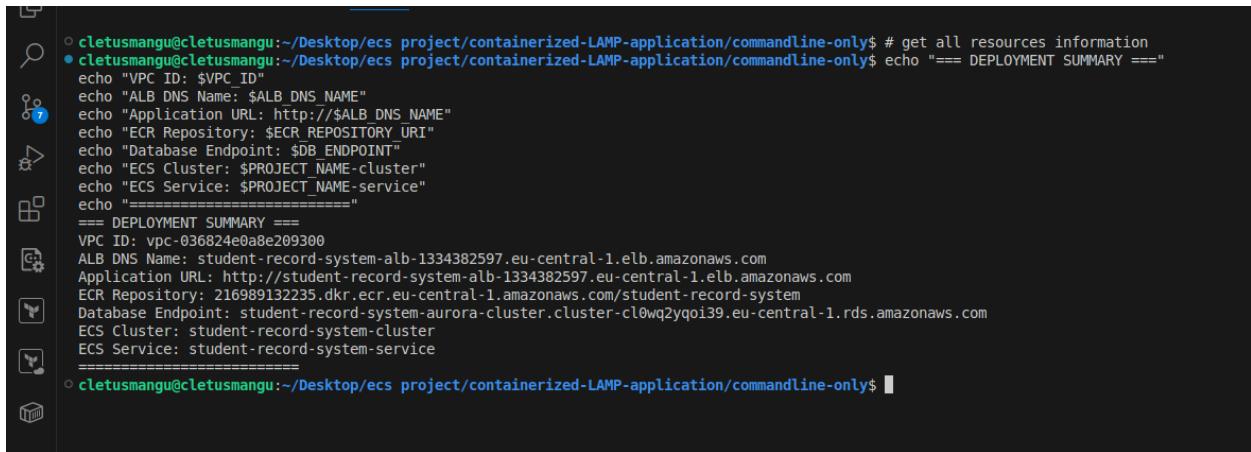
```

● cletusmangu@letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ view logs
● cletusmangu@letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "Recent application logs:" | aws logs tail /aws/ecs/$PROJECT_NAME --since 30m
Recent application logs:
2025-07-05T11:46:42.114000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:46:42 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:46:59.907000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:46:59 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:47:12.126000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:47:12 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:47:29.933000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:47:29 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:47:42.130000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:47:42 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:47:59.960000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:47:59 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:48:12.142000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:48:12 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:48:29.990000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:48:29 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:48:42.173000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:48:42 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:49:00.096000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:49:00 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:49:12.189000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:49:12 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:49:30.064000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:49:30 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:49:42.212000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:49:42 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:50:00.101000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:50:00 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:50:09.493000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:50:09 +0000] "GET / HTTP/1.1" 200 405
3 "-" " "
2025-07-05T11:50:12.216000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.2.160 - - [05/Jul/2025:11:50:12 +0000] "GET / HTTP/1.1" 200 135
5 "-" "ELB-HealthChecker/2.0"
2025-07-05T11:50:30.116000+00:00 ecs/student-record-system/b55b120818654353be586549262eac89 10.0.1.155 - - [05/Jul/2025:11:50:30 +0000] "GET / HTTP/1.1" 200 135
● cletusmangu@letusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ 

```

Get All Resource Information

```
echo "==== DEPLOYMENT SUMMARY ==="
echo "VPC ID: $VPC_ID"
echo "ALB DNS Name: $ALB_DNS_NAME"
echo "Application URL: http://\$ALB\_DNS\_NAME"
echo "ECR Repository: $ECR_REPOSITORY_URI"
echo "Database Endpoint: $DB_ENDPOINT"
echo "ECS Cluster: $PROJECT_NAME-cluster"
echo "ECS Service: $PROJECT_NAME-service"
echo "=====
```



The screenshot shows a terminal window with the following content:

```
○ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ # get all resources information
● cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ echo "==== DEPLOYMENT SUMMARY ==="
echo "VPC ID: $VPC_ID"
echo "ALB DNS Name: $ALB_DNS_NAME"
echo "Application URL: http://\$ALB\_DNS\_NAME"
echo "ECR Repository: $ECR_REPOSITORY_URI"
echo "Database Endpoint: $DB_ENDPOINT"
echo "ECS Cluster: $PROJECT_NAME-cluster"
echo "ECS Service: $PROJECT_NAME-service"
echo "====="
==== DEPLOYMENT SUMMARY ===
VPC ID: vpc-036824e0aae209300
ALB DNS Name: student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com
Application URL: http://student-record-system-alb-1334382597.eu-central-1.elb.amazonaws.com
ECR Repository: 216989132235.dkr.ecr.eu-central-1.amazonaws.com/student-record-system
Database Endpoint: student-record-system-aurora-cluster.cluster-cl0wq2yqoi39.eu-central-1.rds.amazonaws.com
ECS Cluster: student-record-system-cluster
ECS Service: student-record-system-service
=====
○ cletusmangu@pletusmangu:~/Desktop/ecs project/containerized-LAMP-application/commandline-only$ █
```

Cleanup Resources

If you want to tear down everything:

```
# Delete ECS Service
aws ecs update-service --cluster "$PROJECT_NAME-cluster" --service "$PROJECT_NAME-service" --desired-count 0
aws ecs wait services-stable --cluster "$PROJECT_NAME-cluster" --services "$PROJECT_NAME-service"
aws ecs delete-service --cluster "$PROJECT_NAME-cluster" --service "$PROJECT_NAME-service"
```

```
# Delete ECS Cluster
aws ecs delete-cluster --cluster "$PROJECT_NAME-cluster"
```

```
# Delete ALB
```

```
aws elbv2 delete-load-balancer --load-balancer-arn $ALB_ARN
aws elbv2 delete-target-group --target-group-arn $TARGET_GROUP_ARN

# Delete RDS
aws rds delete-db-instance --db-instance-identifier "$PROJECT_NAME-aurora-instance-1" --
skip-final-snapshot
aws rds delete-db-cluster --db-cluster-identifier $DB_CLUSTER_ID --skip-final-snapshot

# Delete other resources...
# (Continue with VPC, Security Groups, etc.)
```