# **Moodle 4.5 Installation Guide for AWS with High Availability Setup**

## **Prerequisites**

* Ubuntu 24.04 EC2 instance
* AWS RDS PostgreSQL 16
* AWS Application Load Balancer
* AWS EBS volume
* AWS S3 bucket
* Imperva account
* Domain name with DNS access
* Let's Encrypt SSL certificate

## **1. Initial Server Setup**

### **Update System**

sudo apt update  
sudo apt upgrade -y

### **Install Required Packages**

sudo apt install -y nginx php8.2-fpm php8.2-common php8.2-mysql php8.2-xml php8.2-curl php8.2-gd \  
php8.2-imagick php8.2-cli php8.2-dev php8.2-imap php8.2-mbstring php8.2-soap php8.2-zip php8.2-intl \  
php8.2-pgsql nfs-common

## **2. PostgreSQL RDS Setup**

### **Configure PostgreSQL Client**

sudo apt install postgresql-client-16

### **Create Moodle Database**

psql -h your-rds-endpoint -p 5402 -U postgres  
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;  
CREATE USER moodleuser WITH PASSWORD 'your-secure-password';  
GRANT ALL PRIVILEGES ON DATABASE moodle TO moodleuser;

## **3. EBS and NFS Setup**

### **Mount EBS Volume**

sudo mkdir /mnt/moodledata  
sudo mount /dev/xvdf /mnt/moodledata

### **Configure NFS Server**

sudo apt install nfs-kernel-server  
echo "/mnt/moodledata \*(rw,sync,no\_subtree\_check)" | sudo tee -a /etc/exports  
sudo exportfs -a  
sudo systemctl restart nfs-kernel-server

### **Configure NFS Client (on other EC2 instances)**

sudo mkdir /mnt/moodledata  
sudo mount -t nfs primary-ec2-private-ip:/mnt/moodledata /mnt/moodledata

## **4. S3 Integration**

### **Install AWS CLI**

sudo apt install awscli

### **Create IAM Role for EC2**

* Create an IAM role with S3 access
* Attach the role to EC2 instances

### **Configure S3 in Moodle**

Add to config.php:

$CFG->alternative\_file\_system\_class = '\tool\_s3\file\_system';  
$CFG->s3\_key = 'your-access-key';  
$CFG->s3\_secret = 'your-secret-key';  
$CFG->s3\_bucket = 'your-bucket-name';  
$CFG->s3\_region = 'your-region';

## **5. NGINX Configuration**

### **Create NGINX Config**

server {  
 listen 8080;  
 server\_name your-domain.com;  
 root /var/www/html/moodle;  
   
 client\_max\_body\_size 100M;  
   
 location / {  
 try\_files $uri $uri/ /index.php?$query\_string;  
 }  
   
 location ~ [^/]\.php(/|$) {  
 fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;  
 fastcgi\_pass unix:/var/run/php/php8.2-fpm.sock;  
 fastcgi\_index index.php;  
 include fastcgi\_params;  
 fastcgi\_param PATH\_INFO $fastcgi\_path\_info;  
 fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  
 }  
}

## **6. SSL Configuration with Let's Encrypt**

### **Install Certbot**

sudo apt install certbot python3-certbot-nginx

### **Generate Certificate**

sudo certbot --nginx -d your-domain.com

## **7. Load Balancer Setup**

1. Create Application Load Balancer
   1. Configure listeners for port 443
   2. Add target group with EC2 instances
   3. Set health check path to /login/index.php
2. Update DNS to point to ALB

## **8. Imperva Configuration**

1. Add site to Imperva
2. Configure CDN settings
3. Update DNS to point to Imperva

## **9. Install Moodle**

cd /var/www/html  
git clone git://git.moodle.org/moodle.git  
cd moodle  
git branch -a  
git checkout MOODLE\_45\_STABLE  
  
sudo chmod -R 755 /var/www/html/moodle  
sudo chown -R www-data:www-data /var/www/html/moodle

## **10. Configure Moodle**

Create config.php with:

<?php  
$CFG->dbtype = 'pgsql';  
$CFG->dblibrary = 'native';  
$CFG->dbhost = 'your-rds-endpoint';  
$CFG->dbname = 'moodle';  
$CFG->dbuser = 'moodleuser';  
$CFG->dbpass = 'your-password';  
$CFG->dbport = '5402';  
$CFG->prefix = 'mdl\_';  
  
$CFG->wwwroot = '<https://your-domain.com>';  
$CFG->dataroot = '/mnt/moodledata';  
$CFG->admin = 'admin';  
  
$CFG->directorypermissions = 0777;

## **11. Auto Scaling Configuration**

1. Create Launch Template
   1. Include UserData script for NFS mount
   2. Attach IAM role for S3 access
   3. Include necessary security groups
2. Create Auto Scaling Group
   1. Set desired, minimum, and maximum instances
   2. Configure scaling policies based on CPU/memory usage

## **Security Considerations**

1. Configure Security Groups:
   1. ALB: Allow 80/443 from Imperva IPs
   2. EC2: Allow 8080 from ALB
   3. RDS: Allow 5402 from EC2 security group
   4. NFS: Allow 2049 within EC2 security group
2. Enable WAF on ALB (optional)

## **Monitoring Setup**

1. Configure CloudWatch for:
   1. EC2 metrics
   2. RDS metrics
   3. ALB metrics
   4. S3 metrics
2. Set up alarms for:
   1. High CPU usage
   2. High memory usage
   3. Database connections
   4. Error rates

## **Backup Strategy**

1. Configure RDS automated backups
2. Set up S3 lifecycle policies
3. Create AMI backups of EC2 instances
4. Enable EBS snapshots