**Installing Moodle 4.5 on Amazon Linux 2023 with NGINX and PostgreSQL**

**Prerequisites**

* Amazon Linux 2023 instance with root/sudo access
* Domain name pointing to your server (optional but recommended)

**1. System Update and Required Packages**

# Update system

sudo dnf update -y

# Install required packages

sudo dnf install -y nginx php php-fpm php-cli php-common php-pgsql php-curl \

php-zip php-gd php-intl php-xmlrpc php-soap php-mbstring php-xml \

php-opcache postgresql postgresql-server postgresql-contrib git \

php-pear php-ldap php-pecl-zstd

# Start and enable NGINX and PHP-FPM

sudo systemctl start nginx

sudo systemctl enable nginx

sudo systemctl start php-fpm

sudo systemctl enable php-fpm

**2. Configure PHP**

# Edit php.ini

sudo vi /etc/php.ini

# Add/modify these values:

memory\_limit = 512M

post\_max\_size = 50M

upload\_max\_filesize = 50M

max\_execution\_time = 600

max\_input\_vars = 5000

**3. Download and Configure Moodle**

# Create Moodle directory

sudo mkdir -p /var/www/moodle

cd /var/www

# Download Moodle

sudo git clone -b MOODLE\_45\_STABLE git://git.moodle.org/moodle.git

# Create moodledata directory

sudo mkdir /var/moodledata

sudo chown -R nginx:nginx /var/moodledata

sudo chmod 777 /var/moodledata

# Set permissions

sudo chown -R nginx:nginx /var/www/moodle

**4. Configure NGINX**

# Create Moodle NGINX configuration

sudo vi /etc/nginx/conf.d/moodle.conf

server {

listen 8080; # Listen on 8080 for ALB health checks and traffic

server\_name aprilgroup.rizaki.my.id;

root /var/www/html/moodle;

index index.php index.html index.htm;

# Add this to properly handle ALB headers

set\_real\_ip\_from 172.20.0.0/16; # VPC CIDR

real\_ip\_header X-Forwarded-For;

# Add this for ALB health checks

location /health.php {

access\_log off;

return 200 'OK';

add\_header Content-Type text/plain;

}

location / {

# Add these headers for SSL behind ALB

proxy\_set\_header X-Forwarded-Proto $http\_x\_forwarded\_proto;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header Host $http\_host;

try\_files $uri $uri/ /index.php?$query\_string;

}

location ~ [^/]\.php(/|$) {

fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

fastcgi\_pass unix:/run/php-fpm/www.sock;

fastcgi\_index index.php;

include fastcgi\_params;

# Add these for proper SSL handling behind ALB

fastcgi\_param HTTPS on;

fastcgi\_param HTTP\_X\_FORWARDED\_PROTO https;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_param PATH\_INFO $fastcgi\_path\_info;

# Add these for better performance

fastcgi\_buffer\_size 128k;

fastcgi\_buffers 4 256k;

fastcgi\_busy\_buffers\_size 256k;

fastcgi\_read\_timeout 300;

}

client\_max\_body\_size 100M;

}

Handle preflight OPTIONS request

**5. SELinux Configuration (if enabled)**

# Set correct SELinux contexts

sudo semanage fcontext -a -t httpd\_sys\_content\_t "/var/www/moodle(/.\*)?"

sudo semanage fcontext -a -t httpd\_sys\_rw\_content\_t "/var/moodledata(/.\*)?"

sudo restorecon -R /var/www/moodle

sudo restorecon -R /var/moodledata

# Allow NGINX to connect to PostgreSQL

sudo setsebool -P httpd\_can\_network\_connect\_db 1

**6. Config.php**

<?php // Moodle configuration file

unset($CFG);

global $CFG;

$CFG = new stdClass();

$CFG->dbtype = 'pgsql';

$CFG->dblibrary = 'native';

$CFG->dbhost = 'aws-lab-lms-db.cz8oucw2cafw.ap-southeast-3.rds.amazonaws.com';

$CFG->dbname = 'moodle';

$CFG->dbuser = 'moodleuser';

$CFG->dbpass = 'pa55word';

$CFG->prefix = 'mdl\_';

$CFG->dboptions = array (

'dbpersist' => 0,

'dbport' => 5402,

'dbsocket' => '',

);

//Redis Configuration

$CFG->session\_handler\_class = '\core\session\redis';

$CFG->session\_redis\_host = '127.0.0.1';

$CFG->session\_redis\_port = 6379; // pastikan ini tidak dikomentari

$CFG->session\_redis\_database = 0;

//$CFG->session\_redis\_auth = 'moodler3d!s';

// Menggunakan Redis untuk cache umum

$CFG->cache\_store = 'redis-stats';

//$CFG->wwwroot = 'http://43.218.93.131';

$CFG->wwwroot = 'https://aprilgroup.rizaki.my.id';

//$CFG->dataroot = '/mnt/moodledata'';

$CFG->dataroot = '/mnt/moodledata';

//$CFG->cachedir = '/var/cache/moodle';

//$CFG->tempdir = '/var/tmp/moodle';

$CFG->admin = 'admin';

$CFG->sslproxy = true;

$CFG->reverse\_proxy = true; // Add this for ALB

$CFG->getremoteaddrconf = 0; // Add this to properly handle client IPs

//Objectfs

$CFG->alternative\_file\_system\_class = '\tool\_objectfs\s3\_file\_system';

//=========================================================================

// 7. SETTINGS FOR DEVELOPMENT SERVERS - not intended for production use!!!

//=========================================================================

//

// Force a debugging mode regardless the settings in the site administration

@error\_reporting(E\_ALL | E\_STRICT); // NOT FOR PRODUCTION SERVERS!

@ini\_set('display\_errors', '1'); // NOT FOR PRODUCTION SERVERS!

$CFG->debug = (E\_ALL | E\_STRICT); // === DEBUG\_DEVELOPER - NOT FOR PRODUCTION SERVERS!

$CFG->debugdisplay = 1; // NOT FOR PRODUCTION SERVERS!

//

// You can specify a comma separated list of user ids that that always see

// debug messages, this overrides the debug flag in $CFG->debug and $CFG->debugdisplay

// for these users only.

// $CFG->debugusers = '2';

$CFG->tool\_generator\_users\_password = 'any123' ; // this will be the password for the users created in next command

header("Access-Control-Allow-Origin: \*");

header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");

header("Access-Control-Allow-Headers: \*");

header("Access-Control-Max-Age: 86400");

require\_once(\_\_DIR\_\_ . '/lib/setup.php');

// There is no php closing tag in this file,

// it is intentional because it prevents trailing whitespace problems!

**7. Install Redis 7**

I can help you install Redis 7 on Amazon Linux 2023. Since it's not available in the default repositories, we'll need to compile it from source. Here's a step-by-step guide:

1. First, update your system and install the necessary dependencies:

sudo dnf update -y

sudo dnf install -y gcc make wget tar

1. Download Redis 7.x source code:

wget https://download.redis.io/releases/redis-7.2.4.tar.gz

1. Extract the archive:

tar xzf redis-7.2.4.tar.gz

cd redis-7.2.4

1. Compile Redis:

make

1. Install Redis:

sudo make install

1. Create directories for Redis configuration and data:

sudo mkdir -p /etc/redis

sudo mkdir -p /var/lib/redis

1. Copy the default configuration file:

sudo cp redis.conf /etc/redis/

1. Edit the configuration file to make necessary adjustments:

sudo nano /etc/redis/redis.conf

Make these changes in the configuration file:

* Set daemonize to yes
* Set supervised to systemd
* Set dir to /var/lib/redis
* Optionally, set a password with requirepass yourpassword

1. Create a systemd service file to manage Redis:

sudo nano /etc/systemd/system/redis.service

Add the following content:

[Unit]

Description=Redis In-Memory Data Store

After=network.target

[Service]

User=redis

Group=redis

ExecStart=/usr/local/bin/redis-server /etc/redis/redis.conf

ExecStop=/usr/local/bin/redis-cli shutdown

Restart=always

[Install]

WantedBy=multi-user.target

1. Create a Redis user:

sudo useradd -r -M -s /bin/false redis

sudo chown redis:redis /var/lib/redis

1. Enable and start the Redis service:

sudo systemctl daemon-reload

sudo systemctl enable redis

sudo systemctl start redis

1. Verify that Redis is running:

sudo systemctl status redis

redis-cli ping

yuaris

psql -h aws-lab-lms-db-rds.cz8oucw2cafw.ap-southeast-3.rds.amazonaws.com -U postgres -d postgres