

## Project: Deploying HA 3-Tier-Architecture (3티어 고가용성 웹앱 구축)

### <Agenda>KOR

- Wordpress 웹에 대한 로드를 분산하기 위해 ALB 사용
- Auto scaling group을 사용해서 로드를 유동성 있게 처리
- 데이터를 저장하기 위한 RDS Aurora 생성
- 서버 데이터의 일관성을 위해 공유 파일 시스템 EFS 사용

### <Agenda>ENG

- Used ALB to distribute the load for Wordpress website
- With Auto scaling group, can handle the load with elasticity
- RDS for saving data being produced from the wordpress website
- For Persistency of datas in the server, used shared file system which is EFS

**Step1.** RDS Aurora 생성 (생성 하기 전 Region이 Seoul로 선택되어있는지 확인)

**Step1.** Create RDS Aurora (Make sure the Region is selected to Seoul)

RDS > Create database

### Create database


Choose a database creation method [Info](#)


☒ **Standard create**  
You set all of the configuration options, including ones for availability, security, backups, and maintenance.


☐ **Easy create**  
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.


### Engine options


Engine type [Info](#)


☒ **Amazon Aurora**  


☐ MySQL  


☐ MariaDB  


☐ PostgreSQL  


☐ Oracle  


☐ Microsoft SQL Server  


Edition

☒ Amazon Aurora MySQL-Compatible Edition

☐ Amazon Aurora PostgreSQL-Compatible Edition

### Engine version [Info](#)

View the engine versions that support the following database features.

#### ▼ Hide filters

- ☐ **Show versions that support the global database feature**  
Allows a single Amazon Aurora database to span multiple AWS Regions.
- ☐ **Show versions that support the parallel query feature**  
Improves the performance of analytic queries by pushing processing down to the Aurora storage layer.
- ☐ **Show versions that support Serverless v2**  
Offers instance scaling for even the most demanding workloads.

Available versions (36/37) [Info](#)

Aurora (MySQL 5.7) 2.10.2 ▼

## Templates

Choose a sample template to meet your use case.

☐ **Production**  
Use defaults for high availability and fast, consistent performance.

☒ **Dev/Test**  
This instance is intended for development use outside of a production environment.

## Settings

### DB cluster identifier [Info](#)

Type a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

Wordpress

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

### ▼ Credentials Settings

#### Master username [Info](#)

Type a login ID for the master user of your DB instance.

admin

1 to 32 alphanumeric characters. First character must be a letter.

- ☐ **Auto generate a password**  
Amazon RDS can generate a password for you, or you can specify your own password.

#### Master password [Info](#)

.....

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

#### Confirm password [Info](#)

.....

## Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

### DB instance class [Info](#)

- ☐ Memory optimized classes (includes r classes)
- ☒ Burstable classes (includes t classes)

db.t3.small

2 vCPUs 2 GiB RAM Network: 2,085 Mbps

- ☐ Include previous generation classes

## Availability & durability

### Multi-AZ deployment [Info](#)

- ☐ Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)  
Creates an Aurora Replica for fast failover and high availability.
- ☒ Don't create an Aurora Replica

## Connectivity [Info](#)



### Compute resource

Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

- ☒ Don't connect to an EC2 compute resource  
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

- ☐ Connect to an EC2 compute resource  
Set up a connection to an EC2 compute resource for this database.

### Virtual private cloud (VPC) [Info](#)

Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.

Default VPC (vpc-0b0bb2e6527d84d7a)

Only VPCs with a corresponding DB subnet group are listed.

- After a database is created, you can't change its VPC.

### DB Subnet group [Info](#)

Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.

default

### Public access [Info](#)

- ☒ Yes  
RDS assigns a public IP address to the cluster. Amazon EC2 instances and other resources outside of the VPC can connect to your cluster. Resources inside the VPC can also connect to the cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.
- ☐ No  
RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.

### VPC security group (firewall) [Info](#)

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

#### ☒ Choose existing

Choose existing VPC security groups

#### ☐ Create new

Create new VPC security group

### Existing VPC security groups

Choose one or more options

default X

No preference

### Availability Zone [Info](#)

No preference

### ► Additional configuration

## Database authentication

### Database authentication options [Info](#)

#### ☒ Password authentication

Authenticates using database passwords.

#### ☐ Password and IAM database authentication

Authenticates using the database password and user credentials through AWS IAM users and roles.

DB가 정상적으로 생성된 것을 확인해줍니다.

***Check if the database is successfully created.***

RDS > Databases

**Databases** Group resources Refresh Modify

Filter by databases

	DB identifier	Role	Engine	Region & AZ	Size	Status
<input type="radio"/>	wordpress	Regional cluster	Aurora MySQL	ap-northeast-2	1 instance	Available
<input type="radio"/>	wordpress-instance-1	Writer Instance	Aurora MySQL	ap-northeast-2b	db.t3.small	Available

3rd Party tool로 rds Instance에 접속(저는 mysql workbench사용하도록 하겠습니다) 접속을 위해서는 Writer Instance의 Endpoint 주소가 필요합니다.

***We'll access the rds Instance using a 3rd party tool(In my case, I'll use mysql workbench, feel free to use other tools) To access the rds Instance, we need Writer Instance's Endpoint.***

RDS > Databases > wordpress

**wordpress** Modify Actions

Related

Filter by databases

	DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance
<input checked="" type="radio"/>	wordpress	Regional cluster	Aurora MySQL	ap-northeast-2	1 instance	Available	-		none
<input type="radio"/>	wordpress-instance-1	Writer Instance	Aurora MySQL	ap-northeast-2b	db.t3.small	Available	8.84%	3 Selects/Sec	none

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance & backups | Tags

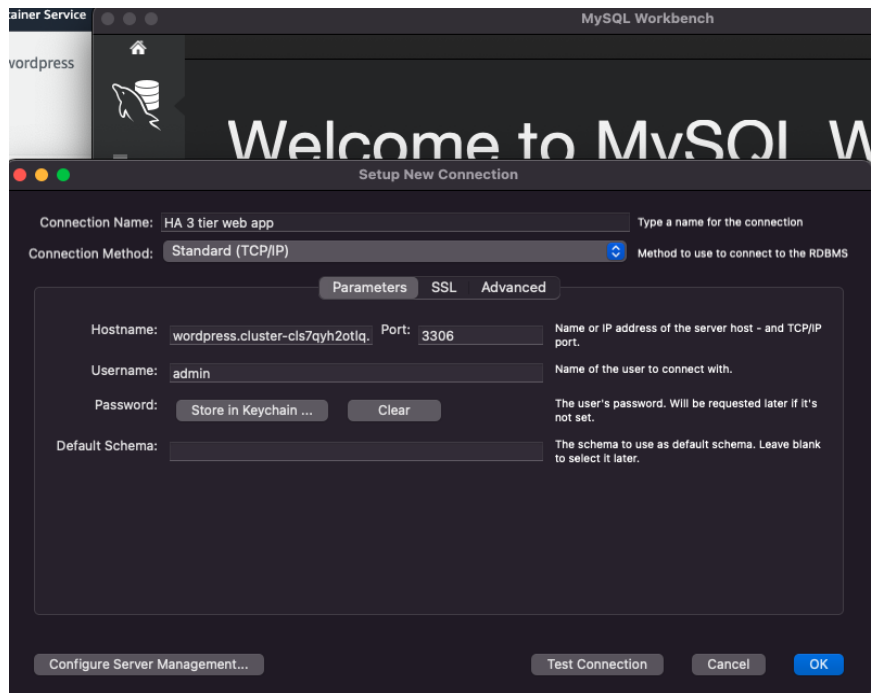
**Endpoints (2)** Actions Create custom endpoint

Filter by endpoint

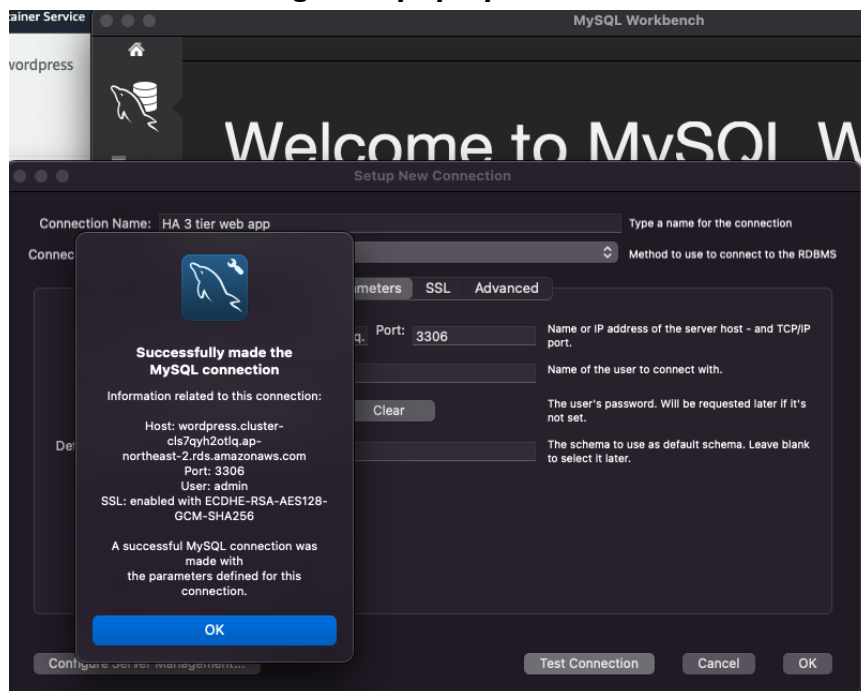
	Endpoint name	Status	Type	Port
<input type="radio"/>	wordpress.cluster-cls7qyh2otlq.ap-northeast-2.rds.amazonaws.com	Available	Writer Instance	3306
<input type="radio"/>	wordpress.cluster-ro-cls7qyh2otlq.ap-northeast-2.rds.amazonaws.com	Available	Reader Instance	3306

Hostname에 Writer Instance의 Endpoint 주소를 넣어주고 포트는 3306에 아이디와 비밀번호는 RDS Aurora 생성하실 때 썼던거 그대로 사용.(Connection name은 원하는 이름 및 Default Schema부분은 공란)

***Use Writer Instance's Endpoint in the Hostname section and 3306 for the port. For Username and Password use what you've configured while creating RDS.(Connection name could be anything that you want and leave Default Schema blank.)***

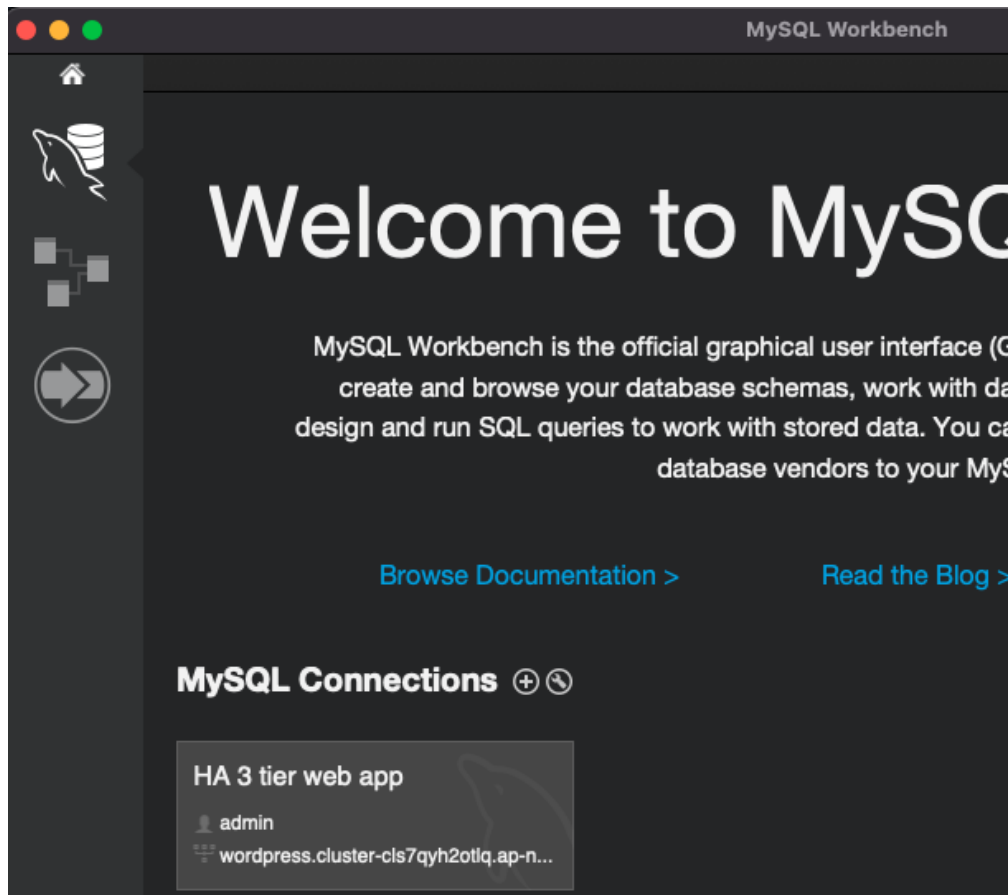


성공적으로 접속되었다면 'Successfully made the MySQL connection' 문구가 발생  
***If you've successfully connected to the DB, 'Successfully made the MySQL connection' message will pop up.***



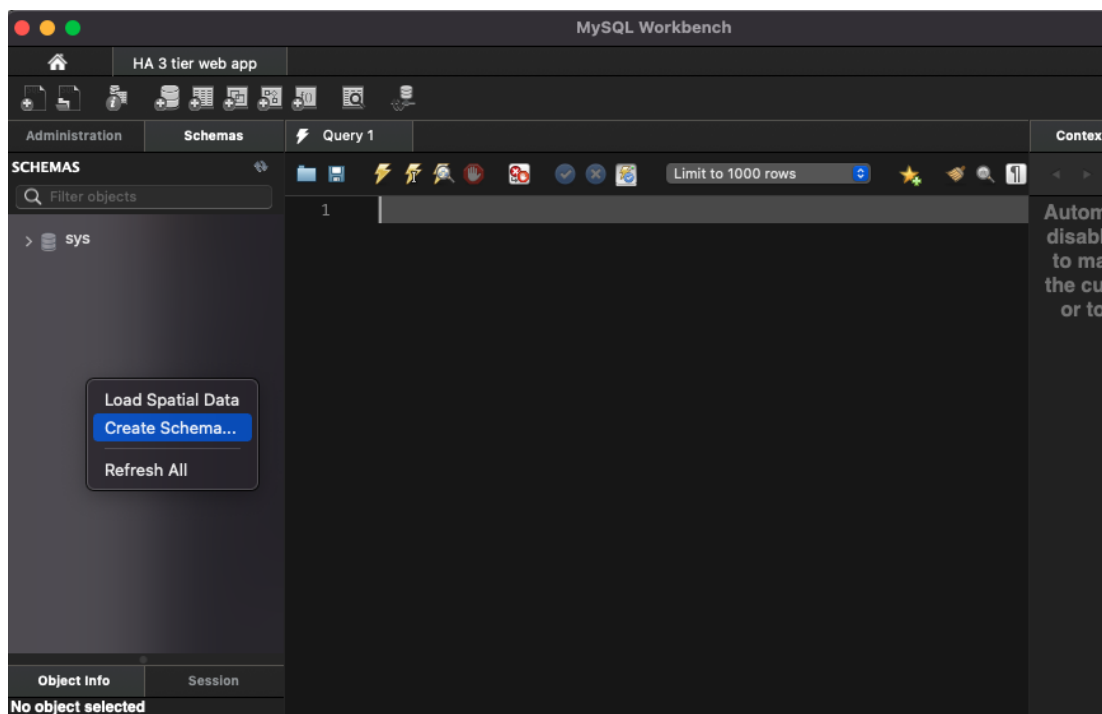
MySQL Connection 밑에 있는 회색 박스 클릭

**Click on the gray box below MySQL Connections**



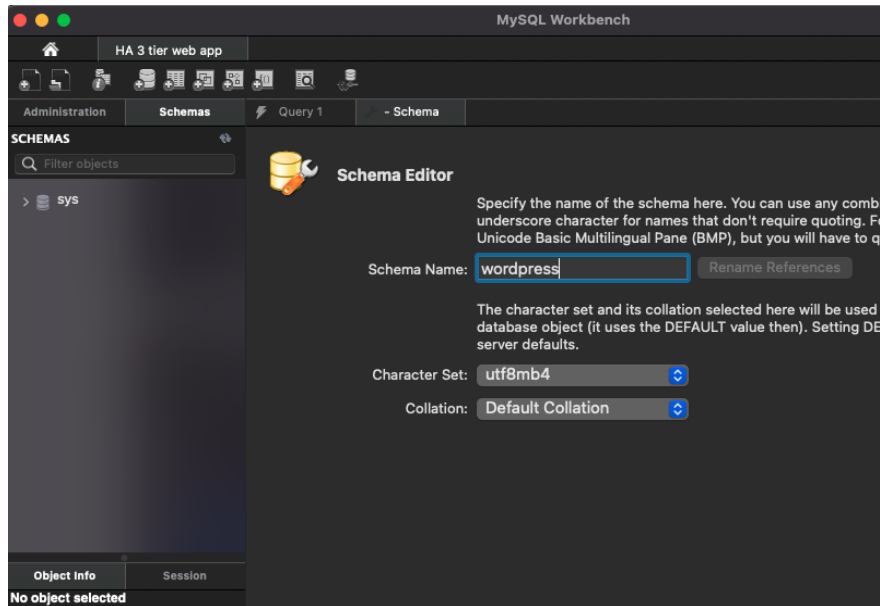
접속 후 Schemas 탭에서 우클릭 후 Create Schema 선택

**After logging into RDS Instance, right click on Schemas tab and select Create Schema**



Schema명은 wordpress 및 Character Set은 utf8mb4로 지정

**Configure Schema name as wordpress and utf8mb4 for Character Set**



**Step2.** 고가용성 데이터 일괄 유지를 위해 EFS생성(생성 하기 전 Region이 Seoul로 선택되어있는지 확인)

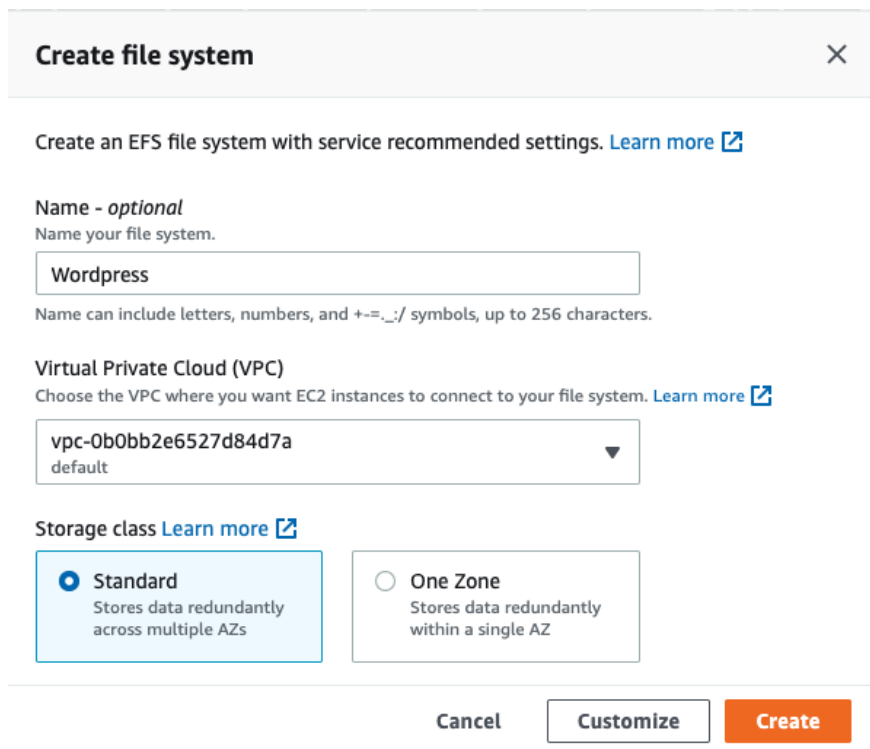
EFS -> Create file system

EFS 이름 지정 후 나머지 값들은 Default 유지

**Step2.** For data persistence in HA environment, will create EFS(Make sure the Region is selected to Seoul)

EFS -> Create file system

After choosing the name for EFS leave everything else to default



EFS 생성된 것 확인

**Double check if EFS is created**

File systems (1)										
Filter by property values										
	Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	File system state	Creation time	Availability Zone
<input type="radio"/>	Wordpress	fs-00ae4cb56edbbc65c	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	-	Available	Wed, 26 Oct 2022 05:00:23 GMT	Standard

**Step3.** Wordpress 사이트를 운영할 EC2 생성(생성 하기 전 Region이 Seoul로 선택되어있는지 확인)

EC2 생성 -> Amazon Linux 2 선택 -> Instance Type: t2.micro 사용

모든 설정은 기본값으로 설정(단, 태그 추가에서 Key: Name, Value: Wordpress 사용)  
추가적으로 보안 그룹 새로 생성(보안 그룹 이름: SG-Wordpress, 포트: SSH 및 HTTP 0.0.0.0/0으로 오픈)

키페어는 원하는 이름으로 생성

**Step3. Create EC2 for hosting Wordpress site(Make sure the Region is selected to Seoul)**

**Create EC2 -> Select Amazon Linux 2 -> Instance Type: t2.micro**

**Leave other settings to default(Except for tag use Key: Name, Value: Wordpress)**

**In addition, create new security group(security group name: SG-Wordpress, port: Open SSH & HTTP to 0.0.0.0/0)**

**Create key pair for accessing EC2(key name of your choice)**

Instances (1/1) Info							
Find Instance by attribute or tag (case-sensitive)							
<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input checked="" type="checkbox"/>	Wordpress	i-05cd372ed8704abf9	Running	t2.micro	2/2 checks passed	No alarms	ap-northeast-2a

터미널 접속 후 해당 명령어 수행(접속은 ec2 instance connect 사용)

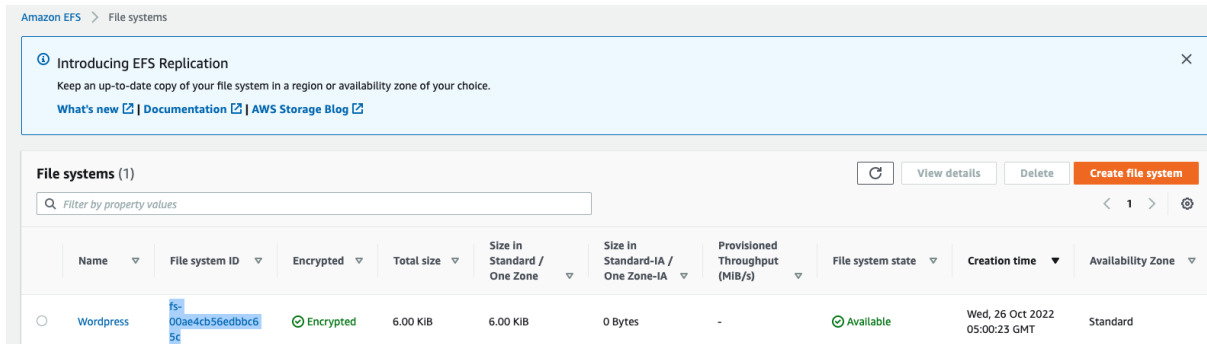
**After logging into terminal, use the commands below(connect to ec2 instance by ec2 instance connect)**

```
[ec2-user@ip-172-31-11-156 ~]$ sudo -s
[root@ip-172-31-11-156 ec2-user]# amazon-linux-extras install -y
lamp-mariadb10.2-php7.2 php7.2
[root@ip-172-31-11-156 ec2-user]# yum -y install httpd
[root@ip-172-31-11-156 ec2-user]# systemctl start httpd
[root@ip-172-31-11-156 ec2-user]# systemctl enable httpd
[root@ip-172-31-11-156 ec2-user]# chown -R ec2-user:ec2-user /var/www/html
[root@ip-172-31-11-156 ec2-user]# chown -R ec2-user:ec2-user /var/www/html
[root@ip-172-31-11-156 ec2-user]# mkdir /var/www/html/wordpress
```



해당 명령어까지 다 수행했으면 EFS로 들어가 파일 시스템ID 복사

**If you're done writing commands mentioned above, go to EFS and check for system ID**



빨간부분의 EFS\_ID를 본인의 EFS 파일 시스템 ID로 변경

**Change the EFS\_ID mentioned in red and replace it with your File system ID**

```
echo "$(curl -s  
http://169.254.169.254/latest/meta-data/placement/availability-zone).[EFS_ID].efs.ap-  
northeast-2.amazonaws.com:/ /var/www/html/wordpress nfs4 defaults" >>  
/etc/fstab
```

EC2로 다시 와 터미널에 아래 명령어 계속 수행

**Come back to EC2 and use the command below in the terminal**

```
[root@ip-172-31-11-156 ec2-user]# echo "$(curl -s  
http://169.254.169.254/latest/meta-data/placement/availability-zone).fs-00ae4cb56ed  
bbc65c.efs.ap-northeast-2.amazonaws.com:/ /var/www/html/wordpress nfs4  
defaults" >> /etc/fstab  
[root@ip-172-31-11-156 ec2-user]# mount -a  
[root@ip-172-31-11-156 ec2-user]# wget https://wordpress.org/latest.tar.gz  
[root@ip-172-31-11-156 ec2-user]# tar -xzf latest.tar.gz  
[root@ip-172-31-11-156 ec2-user]# cp wordpress /var/www/html -r  
[root@ip-172-31-11-156 ec2-user]# chown ec2-user /var/www/html/wordpress  
[root@ip-172-31-11-156 ec2-user]# chmod -R o+r /var/www/html/wordpress
```

압축파일 푼 뒤 wp-config.php 확인(라인 23 ~ 32까지 확인 후 본인의 RDS Instance 값에 맞게 작성) 단 DB\_HOST는 RDS Instance의 라이터 인스턴스 엔드포인트 사용

**Extract the zip file provided for this lab and open wp-config.php(check through like 23 ~ 32 and fill in the value that matches your RDS Instance value) make sure to use RDS Instance's Writer Instance endpoint for DB\_HOST**

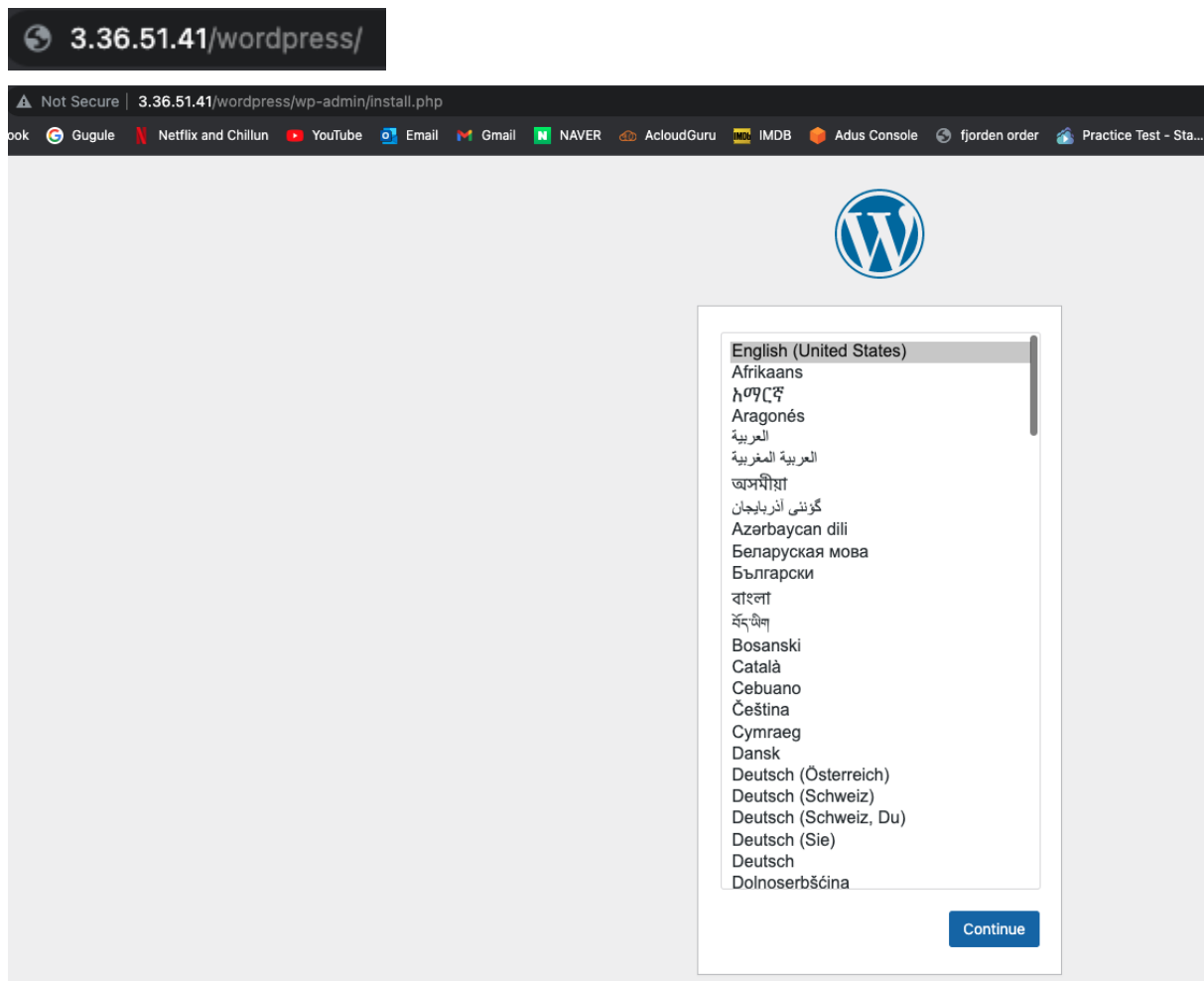
Filezilla나 아무 프로그램 사용해 wp-config.php 파일을 본인의 EC2로 저장

**Use any program like Filezilla to send wp-config.php to your EC2 or in my case I just copied and pasted**

```
[root@ip-172-31-11-156 ec2-user]# vim wp-config.php
[root@ip-172-31-11-156 ec2-user]# ls
latest.tar.gz  wordpress  wp-config.php
[root@ip-172-31-11-156 ec2-user]# cp wp-config.php /var/www/html/wordpress
```


모든 설정들이 알맞게 들어갔다면 Chrome이나 Explorer 주소창에 EC2 public ip/wordpress를 적게되면 아래와 같은 이미지처럼 보인다

***If all configurations are correct, go to Chrome or Explorer and type EC2 public ip/wordpress and you'll see a image like below***



Continue 누르고 아래 이미지와 같이 세팅

**Press continue and fill in the blanks as image shown below**



## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Do not worry, you can always change these settings later.

**Site Title**

**Username**

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

**Password**  [Hide](#)

**Strong**

**Important:** You will need this password to log in. Please store it in a secure location.

**Your Email**

Double-check your email address before continuing.

**Search engine visibility** ☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

[Install WordPress](#)

Site title: RDS Instance 이름(match with RDS Instance name)

Username: RDS Instance 유저명(match with RDS Instance Username)

Password: RDS Instance 비밀번호(match with RDS Instance Password)

Email: 원하는 이메일(your email)

다 세팅 후 Install WordPress 클릭(그리고 다시 EC2 Public IP/wordpress 주소창에 입력)

**After done filling the blanks, press Install WordPress(then use EC2 Public IP/wordpress in chrome or explorer again)**



Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

위와 같은 이미지가 보이면 성공!

If you see an image like this, you've successfully created a wordpress site!

**Step3.** 고가용성 유지를 위해 해당 EC2 Instance에 대한 AMI 생성 후 Auto scaling에 적용(생성 하기 전 **Region**이 **Seoul**로 선택되어있는지 확인)

**Step3.** *For highly available architecture, we'll create an AMI for the wordpress instance we just created and use it with Auto scaling(Make sure the Region is selected to Seoul)*

**Instances (1/1)** [Info](#)

Find instance by attribute or tag (case-sensitive)

Instance state = running X Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Wordpress	i-05cd372ed8704abf9	Running	t2.micro	2/2 checks passed	No alarms	ap-northeast-2a

**Instance: i-05cd372ed8704abf9**

Details Security

Instance summary Info

Instance ID  
i-05cd372ed8704abf9 (Wordpress)

IPv6 address  
-

Launch Instances  
Launch instance from template  
Migrate a server  
Connect  
Stop instance  
Start instance  
Reboot instance  
Hibernate instance  
Terminate instance  
Instance settings  
Networking  
Security  
Image and templates  
Monitor and troubleshoot

Create image  
Create template from instance  
Launch more like this

Status checks Monitoring Tags

Public IPv4 address  
3.36.51.41 | [open address](#)

Private IPv4 addresses  
172.31.11.156

Public IPv4 DNS  
ec2-3-36-51-41.ap-northeast-2a

EC2 > Instances > i-05cd372ed8704abf9 > Create image

**Create image** [Info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID  
i-05cd372ed8704abf9 (Wordpress)

Image name  
Wordpress

Maximum 127 characters. Can't be modified after creation.

Image description - optional  
Image description

Maximum 255 characters

No reboot  
☒ Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/...	Create new snapshot fr...	8	EBS General Purpose S...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together  
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately  
Tag the image and the snapshots with different tags.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Create image

AMI 생성 후 AMI가 Available 상태로 변경됐는지 확인(아래 사진과 같이 Wordpress AMI가 available 상태이면 성공)

**Check if AMI is in Available State after creating it(Like image below, if Wordpress AMI is in available state you're good to go!)**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

### Target group name

Wordpress-TargetGroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

### Protocol

HTTP



### Port

: 80

### VPC

Select the VPC with the instances that you want to include in the target group.

-  
vpc-0b0bb2e6527d84d7a  
IPv4: 172.31.0.0/16



### Protocol version

☒ HTTP1

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ HTTP2

Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

☐ gRPC

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

## Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

### Health check protocol

HTTP



### Health check path

Use the default path of "/" to ping the root, or specify a custom path if preferred.

/wordpress

Up to 1024 characters allowed.

### ▼ Advanced health check settings

Restore defaults

### Port

The port the load balancer uses when performing health checks on targets. The default is the port on which each target receives traffic from the load balancer, but you can specify a different port.

☒ Traffic port

☐ Override

### Healthy threshold

The number of consecutive health checks successes required before considering an unhealthy target healthy.

5

2-10

### Unhealthy threshold

The number of consecutive health check failures required before considering a target unhealthy.

2

2-10

### Timeout

The amount of time, in seconds, during which no response means a failed health check.

seconds

2-120

### Interval

The approximate amount of time between health checks of an individual target

seconds

5-300

### Success codes

The HTTP codes to use when checking for a successful response from a target. You can specify multiple values (for example, "200,202") or a range of values (for example, "200-299").

Wordpress 인스턴스 선택 후 Include as pending below 선택

**Select Wordpress Instance and select Include as pending below**

EC2 > Target groups > Create target group

Step 1  
Specify group details

Step 2  
Register targets

### Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (1/1)

Filter resources by property or value

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Subnet ID
<input checked="" type="checkbox"/>	I-05cd372ed8704abf9	Wordpress	running	SG-Demo-Wordpress	ap-northeast-2a	subnet-0f10d3d7363dab4f0

1 selected

Ports for the selected instances  
Ports for routing traffic to the selected instances.

80

1-65535 (separate multiple ports with commas)

Include as pending below

### Review targets

Targets (1)

Remove all pending

All Filter resources by property or value

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
X	Pending	I-05cd372ed8704abf9	Wordpress	80	running	SG-Demo-Wordpress	ap-northeast-2a	subnet-0f10d3d7363dab4f0

1 pending

Cancel Previous Create target group

EC2 -> Load Balancing -> Load Balancers으로 접속 -> Create Load Balancer 선택

**Go to EC2 -> Load Balancing -> Load Balancers -> Create Load Balancer**



## Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

### ► How Application Load Balancers work

#### Basic configuration

##### Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

##### Scheme [Info](#)

Scheme cannot be changed after the load balancer is created.

☒ Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#) [↗](#)

☐ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

##### IP address type [Info](#)

Select the type of IP addresses that your subnets use.

☒ IPv4

Recommended for internal load balancers.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be confirmed the VPC for your targets, view your [target groups](#).

-

vpc-0b0bb2e6527d84d7a  
IPv4: 172.31.0.0/16

▼

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not available for selection.

☒ ap-northeast-2a

Subnet

subnet-0f10d3d7363dab4f0

▼

IPv4 settings

Assigned by AWS

☒ ap-northeast-2b

Subnet

subnet-0feb2231220e5a72e

▼

IPv4 settings

Assigned by AWS

☒ ap-northeast-2c

Subnet

subnet-08e1aae65624d8790

▼

IPv4 settings

Assigned by AWS

☒ ap-northeast-2d

Subnet

subnet-02d18f6ba914ff641

▼

IPv4 settings

Assigned by AWS


## Security groups [Info](#)


A security group is a set of firewall rules that control the traffic to your load balancer.

### Security groups

Select up to 5 security groups

[Create new security group](#)

default sg-033f269bcb546d72b   
VPC: vpc-0b0bb2e6527d84d7a

SG-Demo-Wordpress sg-0ac7d6055e212d796   
VPC: vpc-0b0bb2e6527d84d7a

## Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

### ▼ Listener HTTP:80

[Remove](#)

Protocol

HTTP

Port

80

1-65535

Default action

[Info](#)

Forward to

Wordpress-TargetGroup

Target type: Instance, IPv4

HTTP

[Create target group](#)

## Summary

Review and confirm your configurations. [Estimate cost](#)

### Basic configuration [Edit](#)

Wordpress-ALB

- Internet-facing
- IPv4

### Security groups [Edit](#)

- default [sg-033f269bcb546d72b](#)
- SG-Demo-Wordpress [sg-0ac7d6055e212d796](#)

### Network mapping [Edit](#)

VPC [vpc-0b0bb2e6527d84d7a](#)

- ap-northeast-2a [subnet-0f10d3d7363dab4f0](#)
- ap-northeast-2b [subnet-0feb2231220e5a72e](#)
- ap-northeast-2c [subnet-08e1aae65624d8790](#)
- ap-northeast-2d [subnet-02d18f6ba914ff641](#)

### Listeners and routing [Edit](#)

- HTTP:80 defaults to [Wordpress-TargetGroup](#)

### Add-on services [Edit](#)

None

### Tags [Edit](#)

None

로드밸런서 생성!

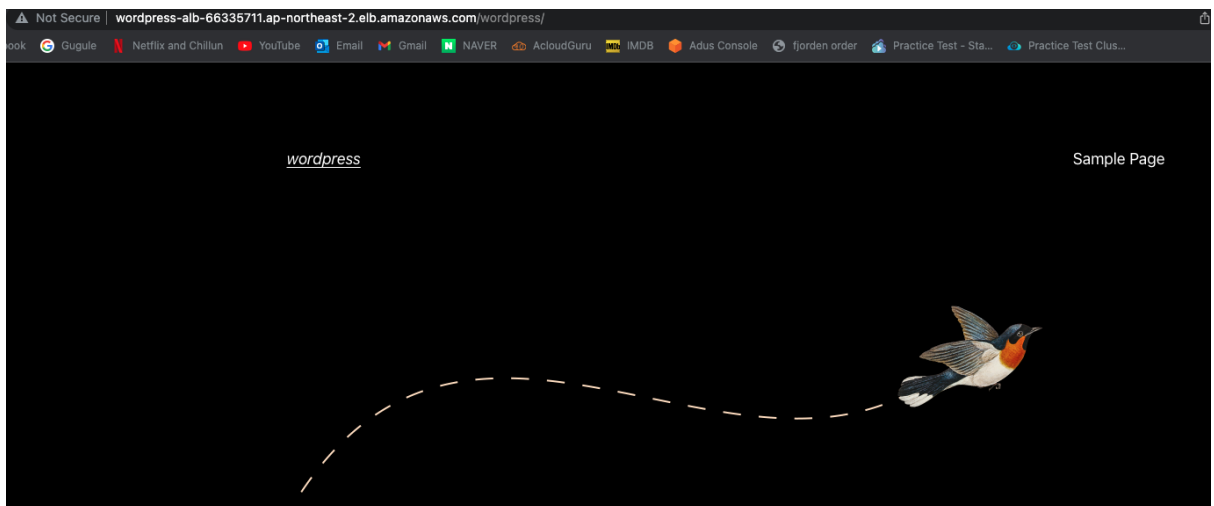
**Create a Load Balancer!**

로드밸런서가 활성화 되면 Load Balancer의 DNS 주소로 접속(Load Balancer 주소/wordpress)

When load balancer becomes active, use Load Balancer's DNS address to access the wordpress site(Load Balancer address/wordpress)

Create Load Balancer Actions					
search : Wordpress-ALB Add filter					
< < 1 to 1 of 1 > >					
Name	DNS name	State	VPC ID	Availability Zones	Type
Wordpress-ALB	Wordpress-ALB-66335711.ap-northeast-2.elb.amazonaws.com	Active	vpc-0b0bb2e6527d84d7a	ap-northeast-2d, ap-nor...	application

Load balancer: Wordpress-ALB	
Description	Listeners Monitoring Integrated services Tags
Basic Configuration	
Name	Wordpress-ALB
ARN	arn:aws:elasticloadbalancing:ap-northeast-2:872858726163:loadbalancer/app/Wordpress-ALB/0042ad13b17a59bb
DNS name	Wordpress-ALB-66335711.ap-northeast-2.elb.amazonaws.com (A Record)
State	Active
Type	application
Scheme	Internet-facing



Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!



**Step4.** 로드 밸런서 생성 후 Auto scaling group에 AMI 저장하기(생성 하기 전 Region이 Seoul로 선택되어있는지 확인)

EC2 -> Auto scaling -> Launch Configurations -> Create Launch Configuration 선택

**Step4.** After creating load balancer, add AMI to the Auto scaling group(Make sure the Region is selected to Seoul)

EC2 -> Auto scaling -> Launch Configurations -> Create Launch Configuration

## Create launch configuration [Info](#)

 Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, [see the documentation](#) 

[Create launch template](#)

### Launch configuration name

Name

### Amazon machine image (AMI) [Info](#)

AMI

### Instance type [Info](#)

Instance type


[Choose instance type](#)

### Additional configuration - optional

Purchasing option [Info](#)☐ Request Spot InstancesIAM instance profile [Info](#)Monitoring [Info](#)☐ Enable EC2 Instance detailed monitoring within CloudWatch

EBS-optimized instance

☐ Launch as EBS-optimized instance[▶ Advanced details](#)

 Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

### Storage (volumes) [Info](#)

#### EBS volumes

[Remove](#)

<input type="checkbox"/>	Volume type	Devices	Snapshot	Size (GiB)	Volume type
<input checked="" type="checkbox"/>	Root	/dev/xvda	snap-083fad596fd64373b	<input type="text" value="8"/>	General purpose SSD (gp2)

[+ Add new volume](#)

 Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

## Security groups [Info](#)

Assign a security group

- ☐ Create a new security group
- ☒ Select an existing security group

### Security groups

[Copy to new](#)[View rules](#)[<](#) [1](#) [2](#) [3](#) [4](#) [>](#)

<input type="checkbox"/>	Security group ID	Name	VPC ID	Description
<input type="checkbox"/>	sg-08aabffc5d53aa211	k8s-elb-aed0e9b25df3f4ee3b7fac4f3d7c9df9	vpc-01595ae19f14cd4aa	Security group for Kubernetes ELB aed0e9b25df3f4ee3b7fac4f3d7c9df9 (default/mario)
<input type="checkbox"/>	sg-09aa77bb83cda0919	wordpress-web-lb-sg	vpc-0b0bb2e6527d84d7a	wordpress-web-lb-sg
<input type="checkbox"/>	sg-09e8ab3ef234640b5	test-vpc-private-sg	vpc-02bb4be92ffed537b	test-vpc-private-sg
<input checked="" type="checkbox"/>	sg-0ac7d6055e212d796	SG-Demo-Wordpress	vpc-0b0bb2e6527d84d7a	SG-Demo-Wordpress

## Key pair (login) [Info](#)

Key pair options

Existing key pair

- ☒ I acknowledge that I have access to the selected private key file (DevEnv.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#)[Create launch configuration](#)

Launch Configuration 생성 완료 후 Auto scaling group 생성

EC2 -> Auto scaling -> Auto scaling Group -> Create Auto scaling group 선택

***After Launch Configuration is created, let's create Auto scaling group***

***EC2 -> Auto scaling -> Auto scaling group -> Create Auto scaling group***

EC2 > Launch configurations > Create launch configuration

Create launch configuration [Info](#)

Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, [see the documentation](#) [↗](#)

Create launch template

Launch configuration name

Name

Wordpress Launch Configuration

Amazon machine image (AMI) [Info](#)

AMI

Wordpress ▼

Instance type [Info](#)

Instance type

t2.micro (1 vCPUs, 1 GiB, EBS Only) 

Choose instance type

Additional configuration - optional

Purchasing option [Info](#)

☐ Request Spot Instances

IAM instance profile [Info](#)

Select IAM role ▼

Monitoring [Info](#)

☐ Enable EC2 instance detailed monitoring within CloudWatch

EBS-optimized instance

☐ Launch as EBS-optimized instance

▶ Advanced details

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

## Storage (volumes) [Info](#)

### EBS volumes

[Remove](#)

<input type="checkbox"/>	Volume type	Devices	Snapshot	Size (GiB)	Volume type
<input type="checkbox"/>	Root	/dev/xvda	snap-083fad596fd64373b	<input type="text" value="8"/>	General purpose SSD (gp2)

[+ Add new volume](#)

 Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

## Security groups [Info](#)

Assign a security group

- ☐ Create a new security group
- ☒ Select an existing security group

### Security groups

[Copy to new](#)[View rules](#)[<](#) [1](#) [2](#) [3](#) [4](#) [>](#)

<input type="checkbox"/>	Security group ID	Name	VPC ID	Description
<input type="checkbox"/>	sg-08aabffc5d53aa211	k8s-elb-aed0e9b25df3f4ee3b7fac4f3d7c9df9	vpc-01595ae19f14cd4aa	Security group for Kubernetes ELB aed0e9b25df3f4ee3b7fac4f3d7c9df9 (default/mario)
<input type="checkbox"/>	sg-09aa77bb83cda0919	wordpress-web-lb-sg	vpc-0b0bb2e6527d84d7a	wordpress-web-lb-sg
<input type="checkbox"/>	sg-09e8ab3ef234640b5	test-vpc-private-sg	vpc-02bb4be92ffed537b	test-vpc-private-sg
<input checked="" type="checkbox"/>	sg-0ac7d6055e212d796	SG-Demo-Wordpress	vpc-0b0bb2e6527d84d7a	SG-Demo-Wordpress

## Key pair (login) [Info](#)

Key pair options

Existing key pair

- ☒ I acknowledge that I have access to the selected private key file (DevEnv.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#)[Create launch configuration](#)



Step 1  
Choose launch template or configuration

Step 2  
Choose instance launch options

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
Add tags

Step 7  
Review

# Choose launch template or configuration [Info](#)

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group. If you currently use launch configurations, you might consider migrating to launch templates.

## Name

### Auto Scaling group name



Enter a name to identify the group.

Wordpress Autoscaler

Must be unique to this account in the current Region and no more than 255 characters.

## Launch configuration [Info](#)

[Switch to launch template](#)

 Instead of using launch configurations to create your EC2 Auto Scaling groups, we recommend that you use launch templates and make use of the Auto Scaling guidance option. For more information on migrating launch configurations and using launch templates, [see the documentation](#) 

### Launch configuration

Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Wordpress Launch Configuration ▼



[Create a launch configuration](#) 

#### Launch configuration

Wordpress Launch Configuration

#### AMI ID

ami-0ebec337bfcfc2550

#### Date created

Wed Oct 26 2022 16:11:16 GMT+0900  
(Korean Standard Time)

#### Security groups

[sg-0ac7d6055e212d796](#) 

#### Instance type

t2.micro

#### Key pair name

DevEnv

Cancel

Next

Step 1

[Choose launch template or configuration](#)

Step 2

**Choose instance launch options**

Step 3 (optional)

[Configure advanced options](#)

Step 4 (optional)

[Configure group size and scaling policies](#)

Step 5 (optional)

[Add notifications](#)

Step 6 (optional)

[Add tags](#)

Step 7

[Review](#)

## Choose instance launch options [Info](#)

Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

### Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

#### VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-0b0bb2e6527d84d7a  
172.31.0.0/16 Default

[Create a VPC](#)

#### Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

ap-northeast-2a | subnet-0f10d3d7363dab4f0  
172.31.0.0/20 Default

ap-northeast-2b | subnet-0feb2231220e5a72e  
172.31.16.0/20 Default

ap-northeast-2c | subnet-08e1aae65624d8790  
172.31.32.0/20 Default

ap-northeast-2d | subnet-02d18f6ba914ff641  
172.31.48.0/20 Default

[Create a subnet](#)[Cancel](#)[Previous](#)[Skip to review](#)[Next](#)

Step 1

[Choose launch template or configuration](#)

Step 2

[Choose instance launch options](#)

Step 3 (optional)

**Configure advanced options**

Step 4 (optional)

[Configure group size and scaling policies](#)

Step 5 (optional)

[Add notifications](#)

Step 6 (optional)

[Add tags](#)

Step 7

[Review](#)

## Configure advanced options [Info](#)

Choose a load balancer to distribute incoming traffic for your application across instances to make it more reliable and easily scalable. You can also set options that give you more control over health check replacements and monitoring.

### Load balancing - optional [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

☒ Attach to an existing load balancer  
Choose from your existing load balancers.

☐ Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

### Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

☒ Choose from your load balancer target groups  
This option allows you to attach Application, Network, or Gateway Load Balancers.

☐ Choose from Classic Load Balancers

#### Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

Wordpress-TargetGroup | HTTP  
Application Load Balancer: Wordpress-ALB

### Health checks - *optional*

#### Health check type [Info](#)

EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

☒ EC2 ☒ ELB

#### Health check grace period

The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.

seconds

### Additional settings - *optional*

#### Monitoring [Info](#)

☐ Enable group metrics collection within CloudWatch

#### Default instance warmup [Info](#)

The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.

☐ Enable default instance warmup

Cancel

Previous

Skip to review

Next

Step 1

[Choose launch template or configuration](#)

Step 2

[Choose instance launch options](#)

Step 3 *(optional)*

[Configure advanced options](#)

Step 4 *(optional)*

**Configure group size and scaling policies**

Step 5 *(optional)*

[Add notifications](#)

Step 6 *(optional)*

[Add tags](#)

Step 7

[Review](#)

## Configure group size and scaling policies [Info](#)

Set the desired, minimum, and maximum capacity of your Auto Scaling group. You can optionally add a scaling policy to dynamically scale the number of instances in the group.

### Group size - *optional* [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

#### Desired capacity

#### Minimum capacity

#### Maximum capacity

### Scaling policies - *optional*

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. [Info](#)

#### ☐ Target tracking scaling policy

Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

#### ☒ None

### Instance scale-in protection - *optional*

#### Instance scale-in protection

If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

☐ Enable instance scale-in protection

Cancel

Previous

Skip to review

Next

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 1  
Choose launch template or configuration

Step 2  
Choose instance launch options

Step 3 (optional)  
Configure advanced options

Step 4 (optional)  
Configure group size and scaling policies

Step 5 (optional)  
Add notifications

Step 6 (optional)  
**Add tags**

Step 7  
-

### Add tags [Info](#)

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

**Tags (1)**

Key: Name, Value - optional: Wordpress-ASG, Tag new instances: ☒, Remove: [Remove]

Add tag, 49 remaining

Cancel Previous Next

**Safely and Securely Replace Instances**  
Configure a maximum amount of time instances in your EC2 Auto Scaling Group can run with Maximum Instance Lifetime.

Wordpress-Autoscaler created successfully

EC2 > Auto Scaling groups

**Auto Scaling groups (1) [Info](#)**

Search your Auto Scaling groups

	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	Wordpress-Autoscaler	Wordpress Launch Configuration	2	-	2	2	2	ap-northeast-2a, ap-northeast-2b, ap-northeast-2c, ap-northeast-2d

아래의 사진과 같이 Auto scaling이 2개의 인스턴스를 더 생성  
***Auto scaling created 2 more wordpress instances automatically***

**Instances (3) [Info](#)**

Find Instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	Wordpress-ASG	i-074a604fa782e9f25	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	ap-northeast-2c	ec2-3-36-132-202.ap-n...
<input type="checkbox"/>	Wordpress	i-05cd372ed8704abf9	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	ap-northeast-2a	ec2-3-36-51-41.ap-nort...
<input type="checkbox"/>	Wordpress-ASG	i-0e95e85c4369343f2	<span>Running</span>	t2.micro	<span>2/2 checks passed</span>	No alarms	ap-northeast-2a	ec2-13-125-106-103.ap...

이렇게 보면 다 완성된 아키텍처같지만 한 가지 치명적인 문제있습니다.

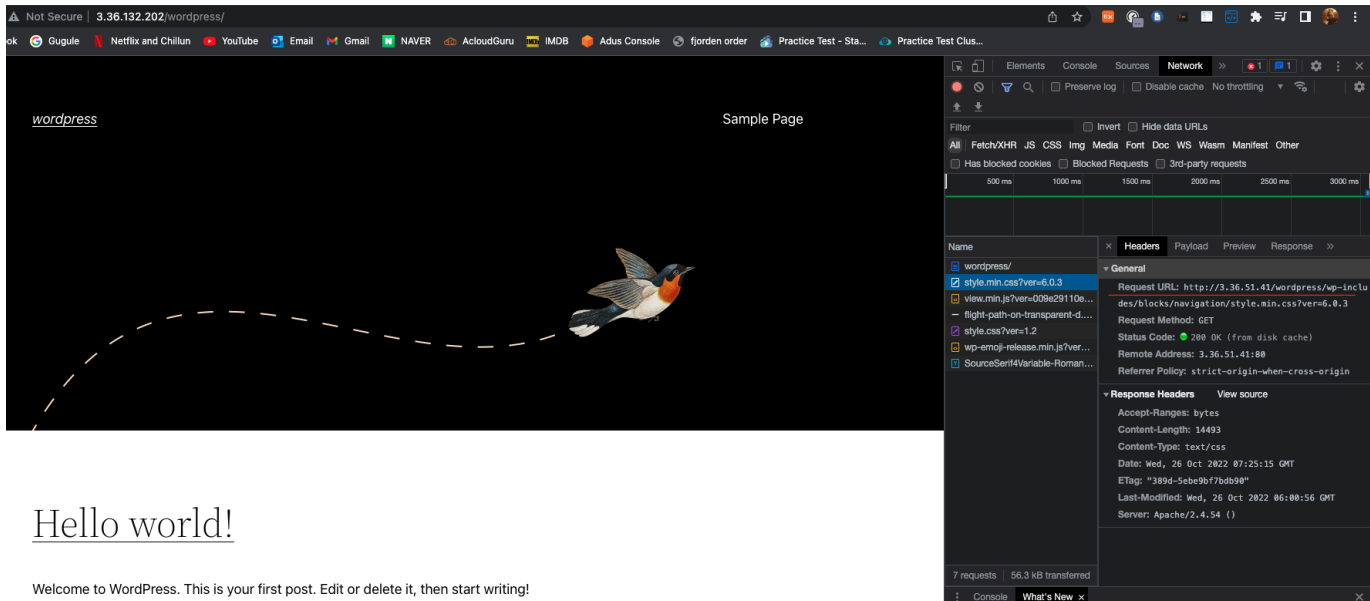
-Autoscaler로 생성된 ec2 주소로 wordpress를 접속해보자

-아래 사진의 빨간줄을 보게되면 주소가 Autoscaler로 생성된 ec2 주소가 아닌 AMI를 생성했던 ec2의 주소임을 알 수 있다.

***You might think it's a completed architecture but there is one crucial problem.***

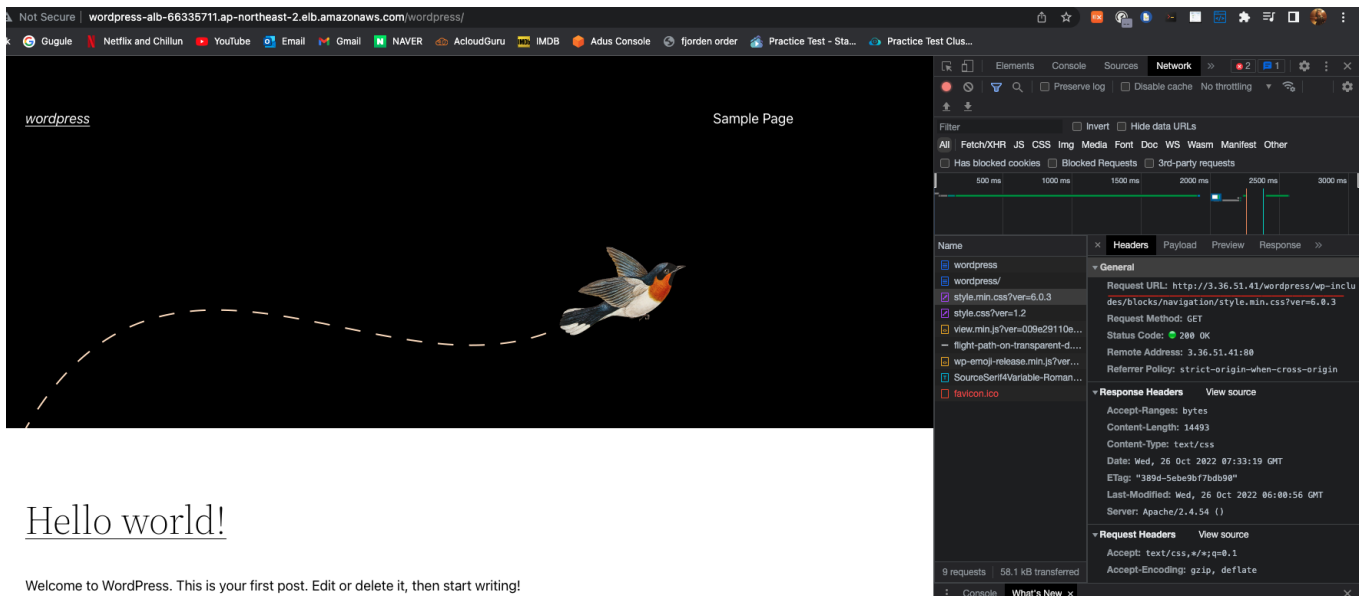
***-Try access a wordpress site that was created through autoscaler***

***-If you see the red line on the photo below, you can see that the ip address is not -from the EC2 that was created with Autoscaler but the original EC2 that we made AMI with.***



-이번에는 로드밸런서 주소로 가져와보자

***Let's try accessing the Wordpress with load balancer's DNS then***



로드밸런서 주소 또한 wordpress AMI를 생성했던 EC2를 가르키고 있다

-이 말 즉슨 만약 AMI를 생성했던 원조 EC2가 종료되면 서비스가 구동을 안한다는 의미이다.

**Load balancer is also pointing to the original EC2 instance that we created ami with**  
**-Which means if the original EC2 gets terminated, wordpress will stop working.**

원조 EC2를 삭제한 후 Autoscaler로 생성된 EC2 주소로 wordpress에 접속해보자  
**Let's try accessing the wordpress site through EC2 that was created with Autoscaler after the original EC2 is terminated.**

×

Successfully terminated i-05cd372ed8704abf9

Instances (3) info

Find instance by attribute or tag (case-sensitive)

Connect

Instance state ▾

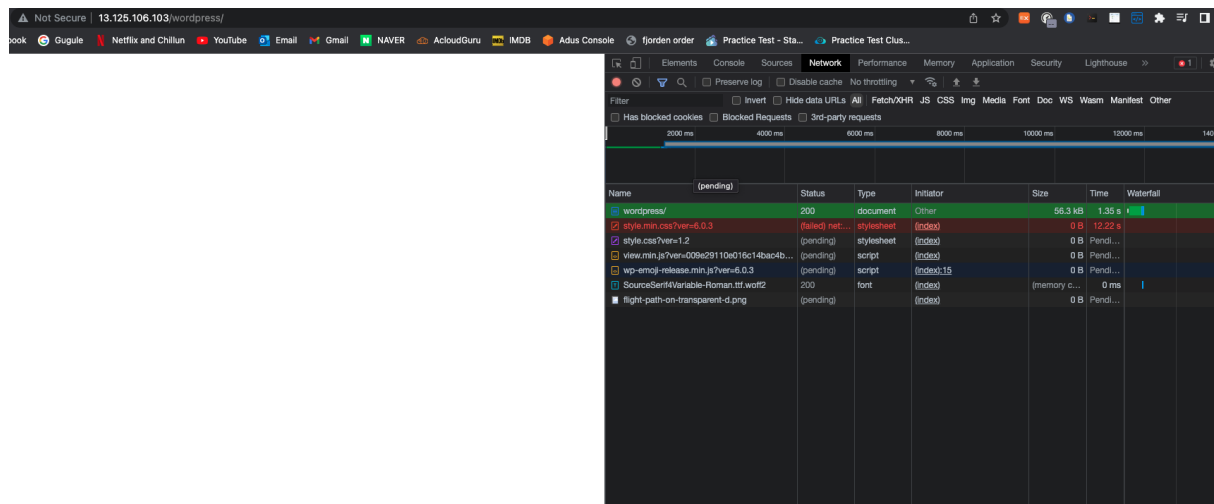
Actions ▾

Launch instances

<input type="checkbox"/>	Name ▾	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone ▾	Public IPv4 DNS	Public IPv4 ... ▾	Elastic IP	IPv6 IPs ▾	Monitoring
<input type="checkbox"/>	Wordpress-ASG	i-074a604fa782e9f25	Running	t2.micro	2/2 checks passed	No alarms +	ap-northeast-2c	ec2-3-36-132-202.ap-n...	3.36.132.202	-	-	disabled
<input type="checkbox"/>	Wordpress	i-05cd372ed8704abf9	Terminated	t2.micro	-	No alarms +	ap-northeast-2a	-	-	-	-	disabled
<input type="checkbox"/>	Wordpress-ASG	i-0e95e85c4369343f2	Running	t2.micro	2/2 checks passed	No alarms +	ap-northeast-2a	ec2-13-125-106-103.ap...	13.125.106.103	-	-	disabled

아래 사진과 같이 이미지가 안뜨거나 error가 발생한 것을 확인할 수 있다.  
그럼 이 문제를 한번 해결해보자

**As image shown below, you can see wordpress is not functioning properly.**  
**Let's try fixing this problem!**



우선 Mysql Workbench로 접속해보자

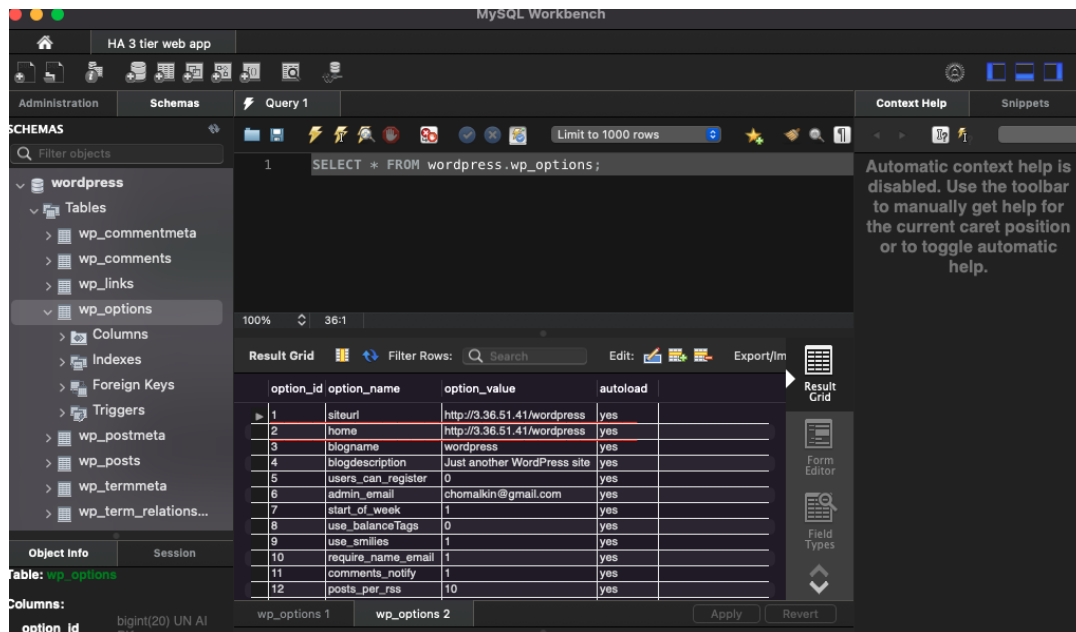
접속 후 `SELECT * FROM wordpress.wp_options;` 명령어를 수행해보자

이후 아래 이미지의 빨간줄 되어있는 부분을 보자

**First access Mysql Workbench**

**Try `SELECT * FROM wordpress.wp_options;`**

**then look at the image shown below where i marked with red line**



그러면 원조 EC2의 주소가 하드코딩 되어있는 것을 확인 할 수 있다  
이거를 해결하기 위한 2가지 방법이 있다.

1. Route 53에 ALB 주소를 등록하고 해당 주소를 하드코딩
2. ALB 주소를 그냥 하드코딩

***As you can see, ip address is hard coded which is original EC2's ip address***

***There are two ways to fix this issue***

1. ***Use Route 53 and assign a record of an ALB and use that address***
2. ***Just use the ALB address that was provided when it was created***



Route 53 > Hosted zones > mdswebservices.com > Create record

Quick create record [Info](#)

Switch to wizard

▼ Record 1

Delete

Record name [Info](#)

Record type [Info](#)

wordpress

.mdswebservices.com

A – Routes traffic to an IPv4 address and some AWS resources

Keep blank to create a record for the root domain.

☒ Alias

Route traffic to [Info](#)

Alias to Application and Classic Load Balancer

Asia Pacific (Seoul) [ap-northeast-2]

Q dualstack.Wordpress-ALB-66335711.ap-northeast-2.elb.amazonaws.com

X

Alias hosted zone ID: ZWKZPGTI48KDX

Routing policy [Info](#)

Evaluate target health

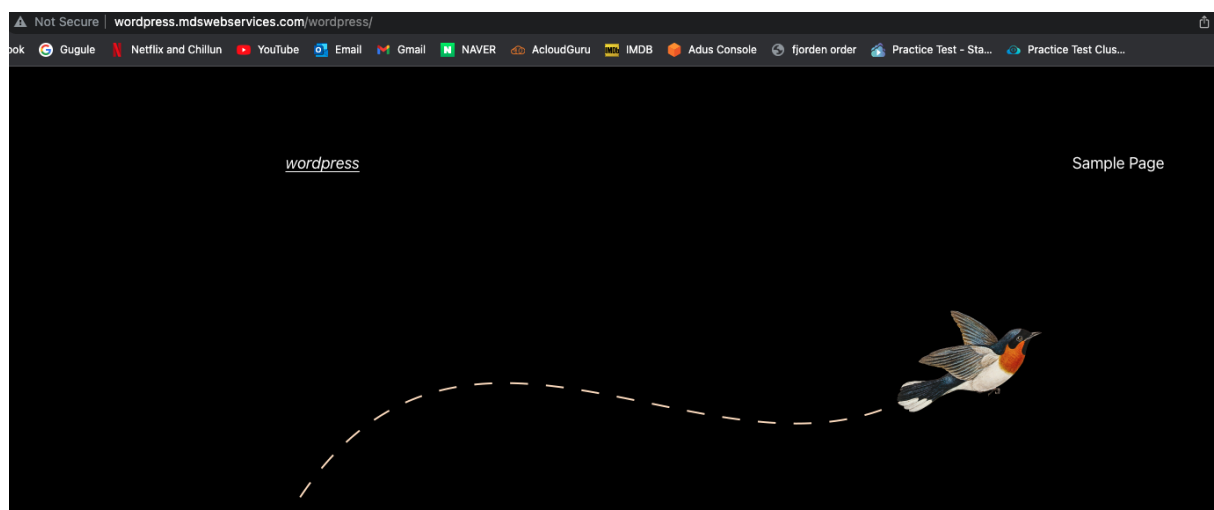
Simple routing

☒ Yes

Add another record

Cancel

Create records



Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!