



ROYAL UNIVERSITY OF PHNOM PENH

DATABASE II

Deploy MySQL and phpmyadmin in AWS EC2 using Docker

Lecturer: Mr. Chhim Bunchhun,



chhim.bunchhun@rupp.edu.kh,



093 222 380



Create sub-domain

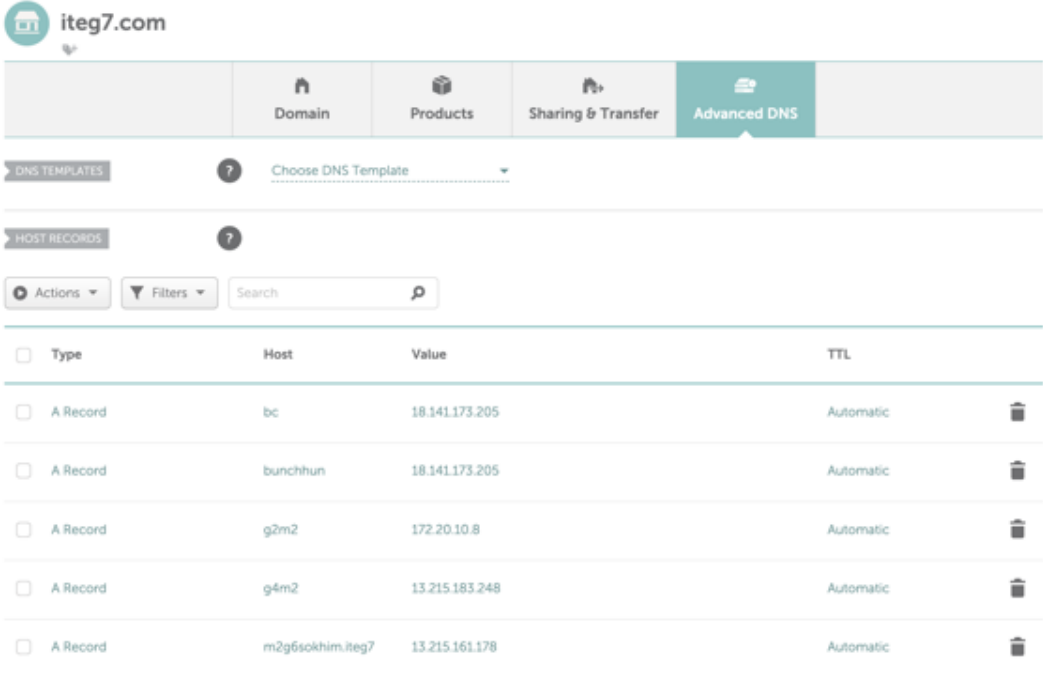


- Go to namecheap.com
- Select domain "iteg7.com" -> Advance DNS
- Create A Record to point to AWS EC2 public IP

Format of subdomain: **m[no]g[no].[member_name]**

Example: m3g1.reaksa

=>We will get: m3g1.reaksa.iteg7.com point to that public IP



Type	Host	Value	TTL
<input type="checkbox"/> A Record	bc	18.141.173.205	Automatic
<input type="checkbox"/> A Record	bunchhun	18.141.173.205	Automatic
<input type="checkbox"/> A Record	g2m2	172.20.10.8	Automatic
<input type="checkbox"/> A Record	g4m2	13.215.183.248	Automatic
<input type="checkbox"/> A Record	m2g6sokhim.iteg7	13.215.161.178	Automatic



- Remote to AWS EC2 server via SSH
- Follow guideline on: <https://docs.docker.com/engine/install/debian/>
- To check if docker installed:

```
docker --version
```

```
root@ip-172-31-26-131:/home/bunchhun# docker --version
Docker version 20.10.18, build b40c2f6
```

Register free account at DockerHub



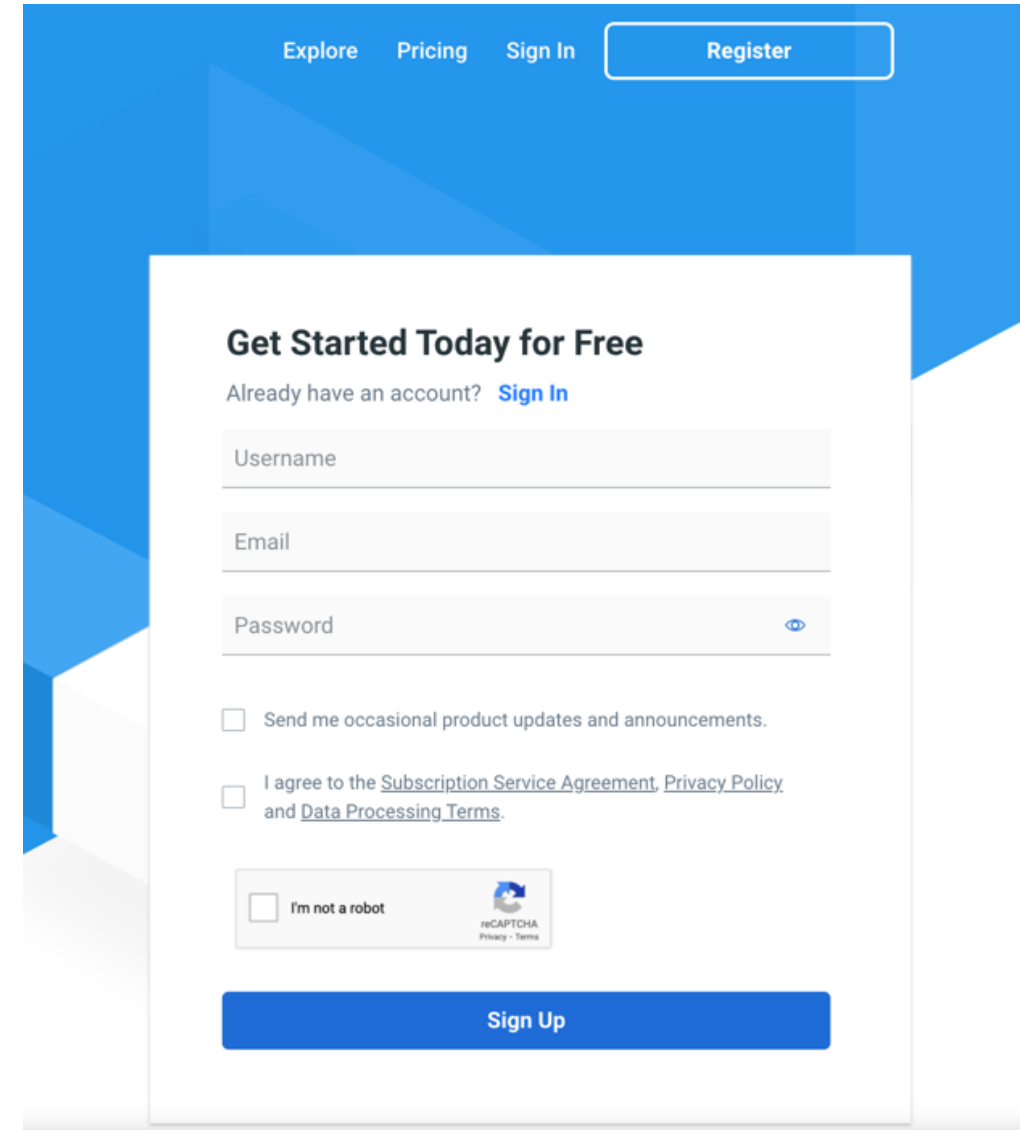
- Link: <https://hub.docker.com>

Docker login

->Provide username and password

```
root@ip-172-31-26-131:/home/bunchhun# docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@ip-172-31-26-131:/home/bunchhun#
```



The screenshot shows the DockerHub registration page. At the top, there are links for 'Explore', 'Pricing', 'Sign In', and a 'Register' button. The main heading is 'Get Started Today for Free'. Below this, there is a link for 'Already have an account? Sign In'. The registration form includes fields for 'Username', 'Email', and 'Password' (with an eye icon for toggling visibility). There are two checkboxes: 'Send me occasional product updates and announcements.' and 'I agree to the Subscription Service Agreement, Privacy Policy and Data Processing Terms.' Below these is a reCAPTCHA widget with the text 'I'm not a robot' and a 'reCAPTCHA Privacy - Terms' link. At the bottom of the form is a large blue 'Sign Up' button.

Pull MySQL and phpmyadmin images from DockerHub



```
docker pull mysql
```

```
docker pull phpmyadmin
```

- To check images that exist:

```
docker image ls
```

 or

```
docker images
```



```
root@ip-172-31-26-131:/home/bunchhun# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
Digest: sha256:b9532b1edea72b6cee12d9f5a78547bd3812ea5db842566e17f8b33291ed2921
Status: Image is up to date for mysql:latest
docker.io/library/mysql:latest
```

```
root@ip-172-31-26-131:/home/bunchhun# docker pull phpmyadmin
Using default tag: latest
latest: Pulling from library/phpmyadmin
Digest: sha256:ea1339b5d1d43d4170eefeec36ab8ac40d076797072e00b1a9625c4d8acbadf5
Status: Image is up to date for phpmyadmin:latest
docker.io/library/phpmyadmin:latest
```

Create Docker Network “mynet”



```
docker network create mynet
```

- To list all existing network

```
docker network ls
```

```
root@ip-172-31-26-131:/home/bunchhun# docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
40b966647bfa        bridge             bridge              local
b816835d4867        host               host                local
161bd70ece68        mynet              bridge              local
1abe650b327a        none              null                local
root@ip-172-31-26-131:/home/bunchhun#
```



- To run MySQL Server Container

```
docker run -d \  
  --name mysql \  
  -p 3306:3306 \  
  -e MYSQL_ROOT_PASSWORD=Rupp4ever! \  
  --network mynet \  
  --restart=always \  
  -it mysql:latest
```

- To check running container: `docker container ls`

```
root@ip-172-31-26-131:/home/bunchhun# docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e023ae157c54	mysql	"docker-entrypoint.s..."	10 hours ago	Up 10 hours	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp	mysql



- Go to instant->select instant->Security->Select id in Security groups

IAM Role

-

Subnet ID

subnet-05becb6eac0251ff4

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Security details

IAM Role

-

Security groups

sg-000b49c0bfa07ea2a (launch-wizard-5)

▼ Inbound rules

Filter rules

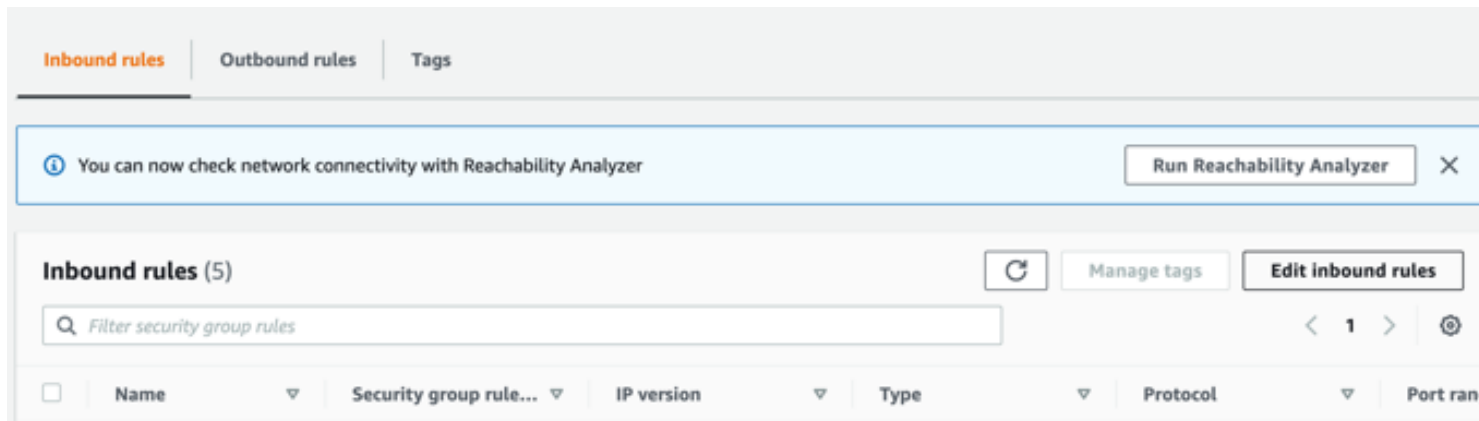
Owner ID

670858105406

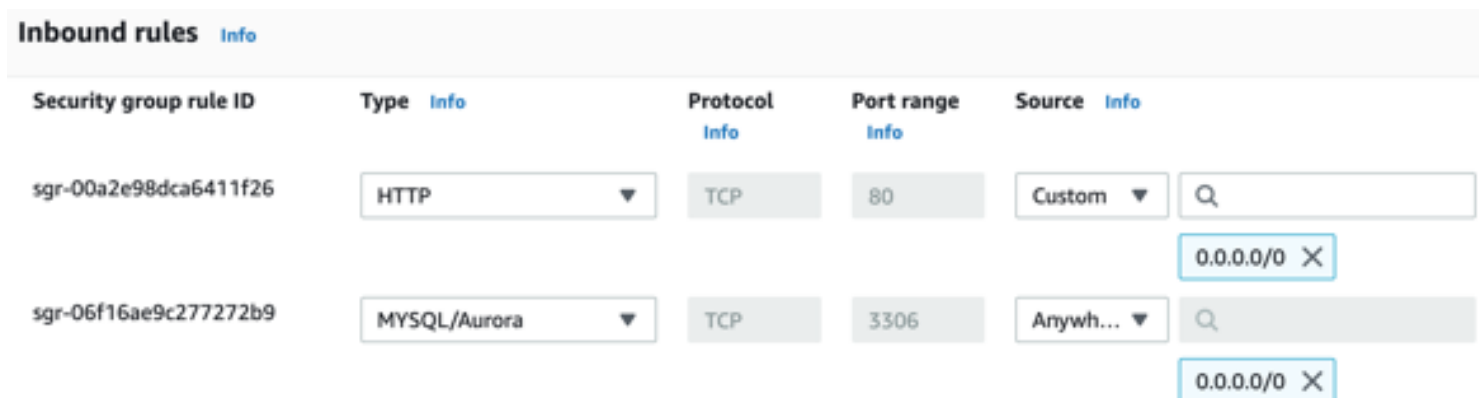
Allow port 3306 in AWS EC2 Firewall

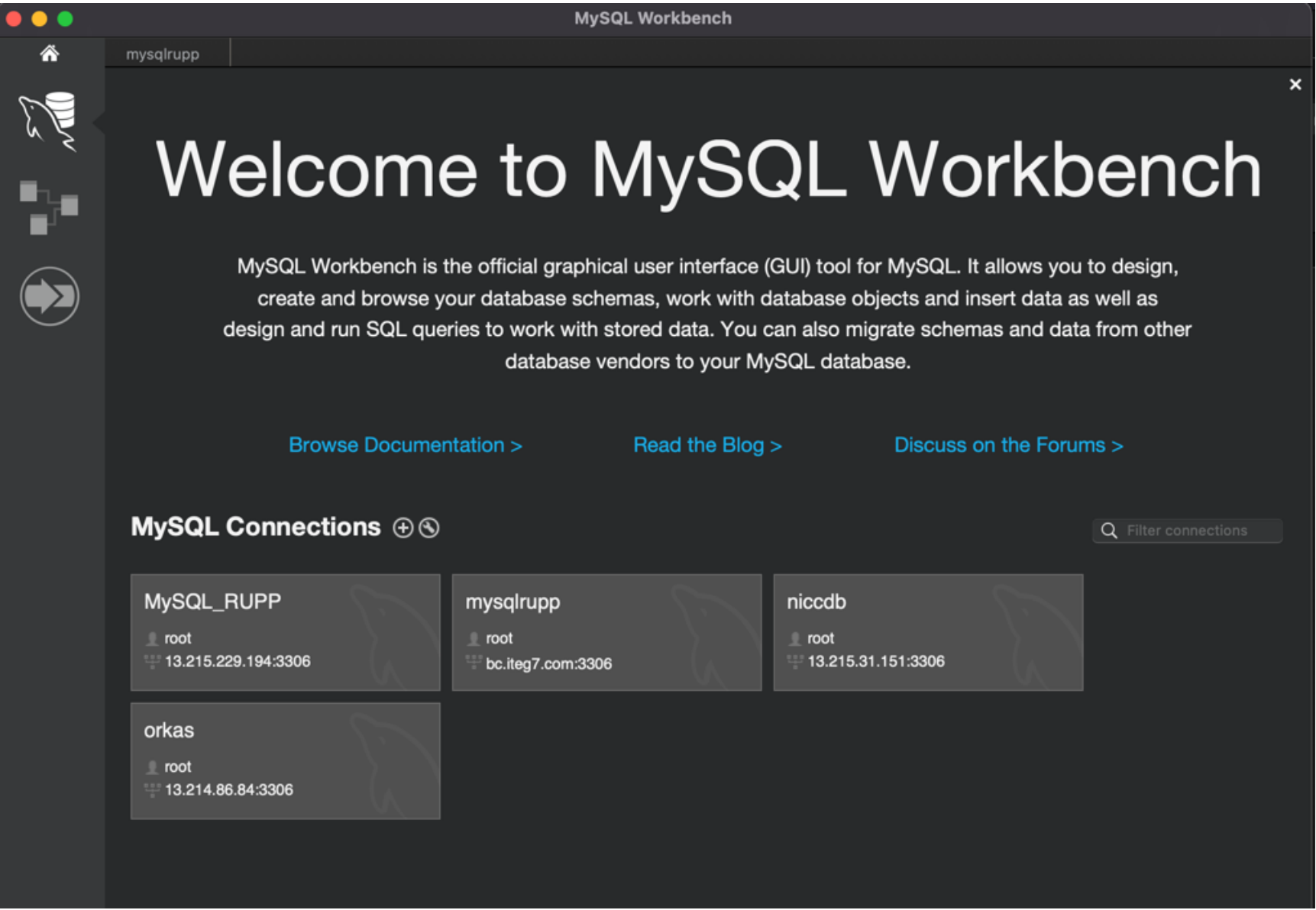


- Click "Edit inbound rules"



- Add rule to allow port 3306:







- To run phpmyadmin container:

```
docker run -d \  
    --name phpmyadmin \  
    -p 8080:80 \  
    -e PMA_HOST=mysql \  
    -e PMA_USER=root \  
    -e PMA_PASSWORD=Rupp4ever! \  
    --network mynet \  
    --restart=always \  
    -it phpmyadmin:latest
```

- To check running container: `docker container ls`

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
53c4210b4f82	phpmyadmin:latest	"/docker-entrypoint..."	11 seconds ago	Up 10 seconds	0.0.0.0:80->80/tcp, :::80->80/tcp
phpmyadmin					
e023ae157c54	mysql	"docker-entrypoint.s..."	10 hours ago	Up 10 hours	0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
mysql					
root@ip-172-31-26-131:/home/bunchhun#					

Allow port 80 and 443 in AWS EC2 Firewall



- Go to instant->select instant->Security->Select id in Security groups

IAM Role

-

Subnet ID

subnet-05becb6eac0251ff4

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Security details

IAM Role

-

Security groups

sg-000b49c0bfa07ea2a (launch-wizard-5)

Owner ID

670858105406

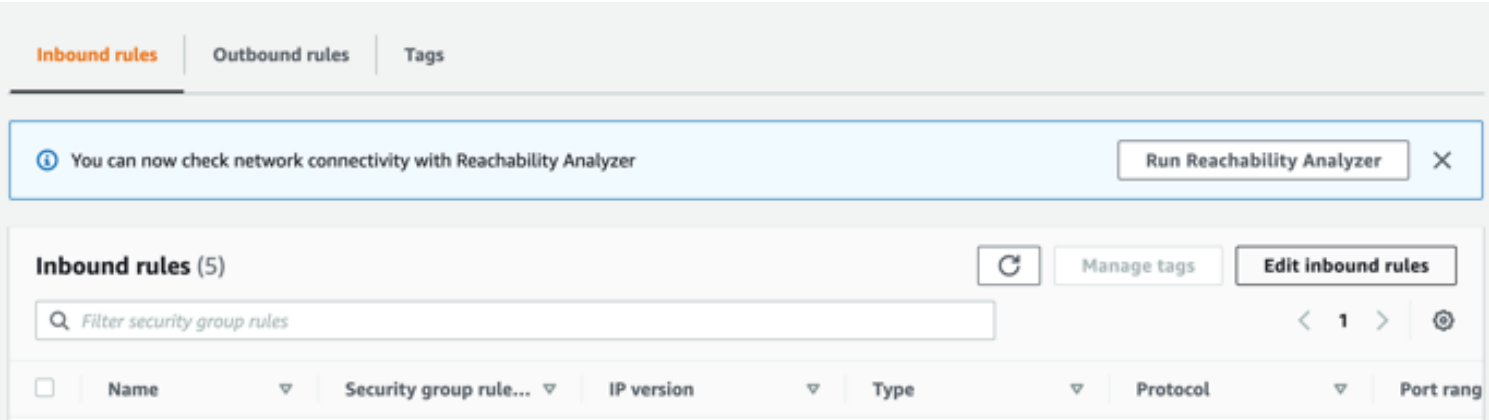
▼ Inbound rules

Filter rules

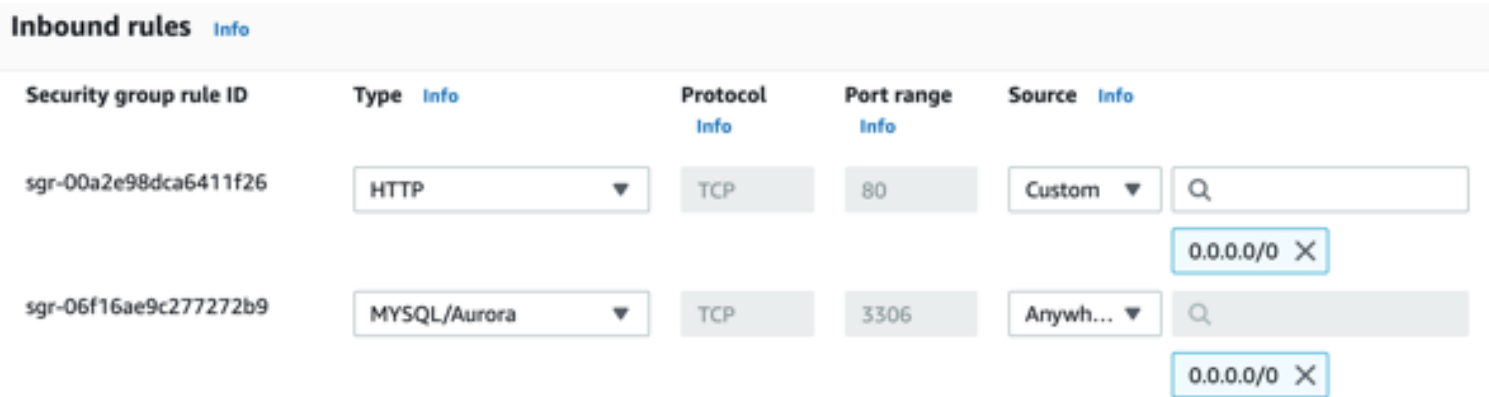
Allow port 80 and 443 in AWS EC2 Firewall



- Click “Edit inbound rules”



- Add rule to allow port 80:





- Install nginx

```
apt install nginx
```

- Install Let's Encrypt for nginx

```
apt install certbot python3-certbot-nginx
```

Configure NGINX with phpmyadmin



- Create new file in /etc/nginx/sites-available

```
nano bc.iteg7.com
```

- o Add below content as on the right

```
bunchhun@ip-172-31-26-131:~$ sudo su
[sudo] password for bunchhun:
root@ip-172-31-26-131:/home/bunchhun# cd /etc/nginx/sites-available/
root@ip-172-31-26-131:/etc/nginx/sites-available# nano bc.iteg7.com
root@ip-172-31-26-131:/etc/nginx/sites-available#
```

- o Activate bc.iteg7.com configuration

```
ln -s /etc/nginx/sites-available/bc.iteg7.com /etc/nginx/sites-enabled
```

- Restart nginx

```
Service nginx restart
```

```
server
{
    listen 80;
    listen [::]:80;
    server_name bc.iteg7.com;
    client_max_body_size 128M;
    access_log /var/log/nginx/access.log;
    error_log /var/log/nginx/error.log;
    location ~ /\.well-known
    {
        allow all;
    }

    location /
    {
        proxy_pass http://127.0.0.1:8080/;
        proxy_http_version 1.1;
        proxy_cache_bypass $http_upgrade;

        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-Host $host;
        proxy_set_header X-Forwarded-Port $server_port;

        proxy_set_header X-Scheme $scheme;
    }
}
```

Get SSL Certificate through Let's Encrypt



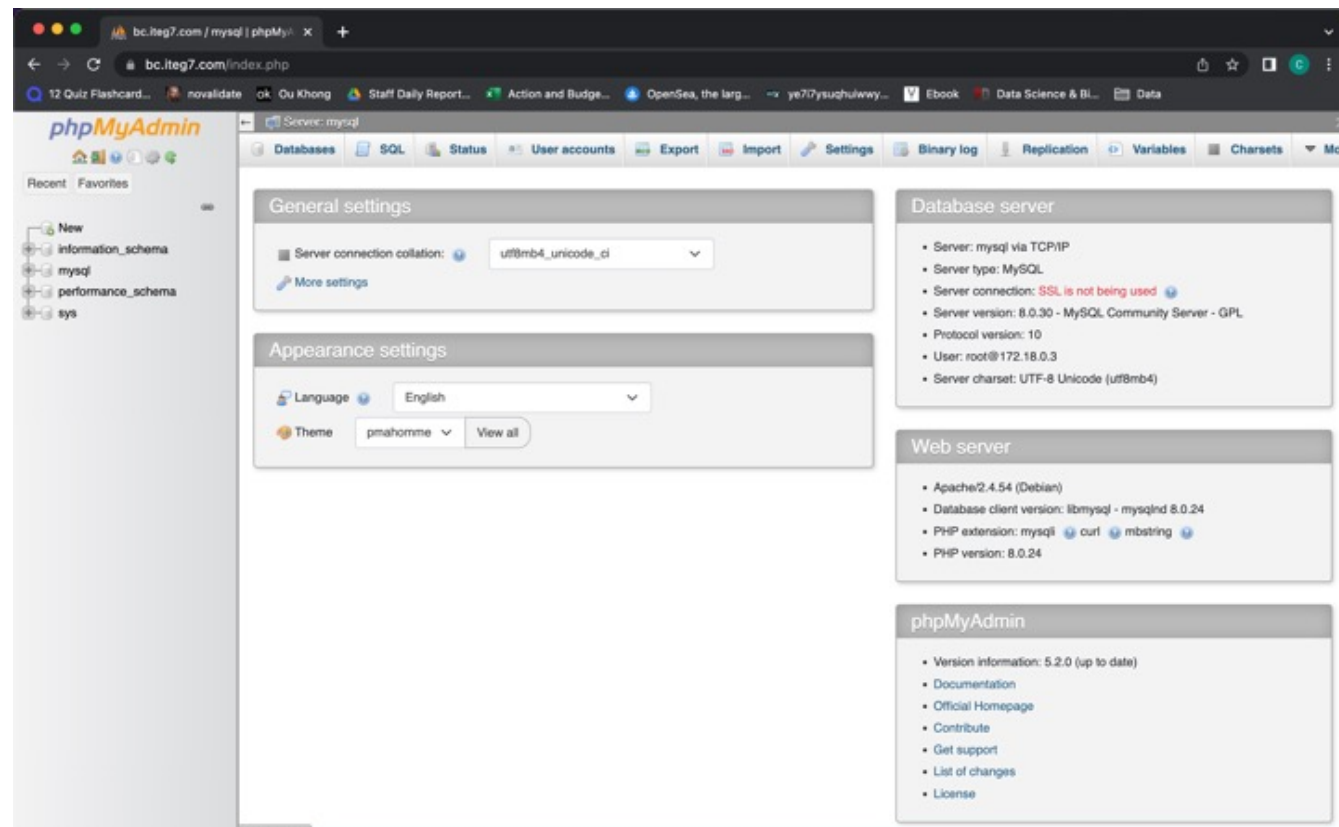
```
certbot --nginx -d bc.iteg7.com
```

```
-> enter email address like :bunchhun@iteg7.com
```

```
-> y
```

⇒ <https://bc.iteg7.com>

working with https





THANK YOU
for your attention !

