

## Run App in Docker using AWS.

Create EC2 instance in AWS.

Create new key pair

Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

AWS-001-key

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA

RSA encrypted private and public key pair

☐ ED25519

ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format

☒ .pem

For use with OpenSSH

☐ .ppk

For use with PuTTY

Cancel

Create key pair

## Network Settings

▼ Network settings Info

Edit

Network Info

vpc-04011abcc910371a9

Subnet Info

No preference (Default subnet in any availability zone)

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere

0.0.0.0/0

☒ Allow HTTPs traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

×

Open Port 8000

Go to Network & Security -> Security Groups

Select security group

## Inbound rules

Edit Inbound rules

Add rule

**Edit inbound rules** [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional	
sg-0c8f65705ce50aad	HTTP	TCP	80	Custom	0.0.0.0	Delete
sg-056d2f118362426b	HTTPS	TCP	443	Custom	0.0.0.0	Delete
sg-0001645d5a7716dc	SSH	TCP	22	Custom	0.0.0.0	Delete
-	Custom TCP	TCP	8000	Anywhere IPv4	0.0.0.0	Delete

[Add rule](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

## Launch Instance

### Connect to instance using SSH client

**Connect to instance** [Info](#)

Connect to your instance i-0b321de7453bbe49d (AWS-001) using any of these options

[EC2 Instance Connect](#) | [Session Manager](#) | **[SSH client](#)** | [EC2 serial console](#)

Instance ID  
[i-0b321de7453bbe49d](#) (AWS-001)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is AWS-001-key.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
`chmod 400 AWS-001-key.pem`
4. Connect to your instance using its Public DNS:  
`ec2-18-212-66-70.compute-1.amazonaws.com`

Example:  
`ssh -i "AWS-001-key.pem" ubuntu@ec2-18-212-66-70.compute-1.amazonaws.com`

**Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

### Step 3

Example `ssh -i "AWS-001-key.pem" ubuntu@ec2-44-201-126-231.compute-1.amazonaws.com`

```
Downloads — ubuntu@ip-172-31-81-204: ~ — ssh -i AWS-001-key.pem ubuntu...

System load: 0.03466796875    Processes:           110
Usage of /:  19.0% of 7.58GB  Users logged in:    0
Memory usage: 21%            IPv4 address for eth0: 172.31.81.204
Swap usage:  0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-81-204:~$
```

## Prepare Instance

### Run in terminal

`sudo apt-get update`

`sudo apt-get upgrade`

`sudo apt-get update`

### Install Docker

`curl -fsSL https://get.docker.com -o get-docker.sh`

`sudo sh get-docker.sh`

`sudo apt-get update`

`sudo reboot`

## Clone git repository

`git clone https://github.com/Luck547/fork4cloud.git`

`git clone https://github.com/Luck547/Mapl-App.git`

`cd fork4cloud/`

`cd Mapl-App/`

`cd app`

## Create docker network

```
sudo docker network create -d bridge mapl-net
```

## Run Mongo

```
sudo docker run -d --network mapl-net -p 27017:27017 -v mapl-vol --name mongodb -e  
MONGO_INITDB_DATABASE='admin' -e MONGO_INITDB_ROOT_USERNAME='root' -e  
MONGO_INITDB_ROOT_PASSWORD='copy&pasteME-547' --label mapl mongo:latest
```

## Build image from dockerfile with tag

```
sudo docker build -t mapl-api .
```

## Run Image

```
sudo docker run -d --network mapl-net -p 8000:8000 -v mapl-vol --name mapl-api --label mapl  
mapl-api
```

## Test App

Find public ip

Run from the browser: <http://44.201.126.231:8000>

## Error

- El fichero main.py no lleva la autenticación

```
client = MongoClient('mongodb://root:copy&pasteME-547@mongodb:27017/')
```

## Docker commands

List docker

```
sudo docker ps -a
```

(Remove docker)

```
sudo docker stop 0ad15bdb65
```

```
sudo docker rm 0ad15bdb65
```

(check docker logs)

```
sudo docker logs 25e4773456e2
```

(run docker container terminal)

(networks)

```
docker network ls
```

(github authentication)

<https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls>