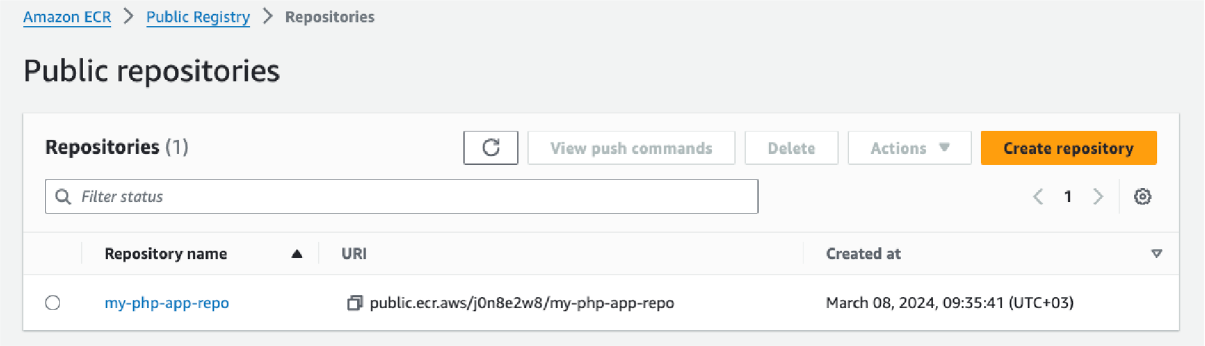
Steps:

1. Create an Amazon ECS Cluster:

1. Open the AWS Management Console and navigate to the Amazon ECS service.
2. Click on "Clusters" in the left navigation pane.
3. Click on "Create cluster".

2. Create an ECR Public Repository:

1. Navigate to the Amazon ECR service in the Management Console.
2. Click on "Repositories" in the left navigation pane.
3. Click on "Create repository".



Public repositories

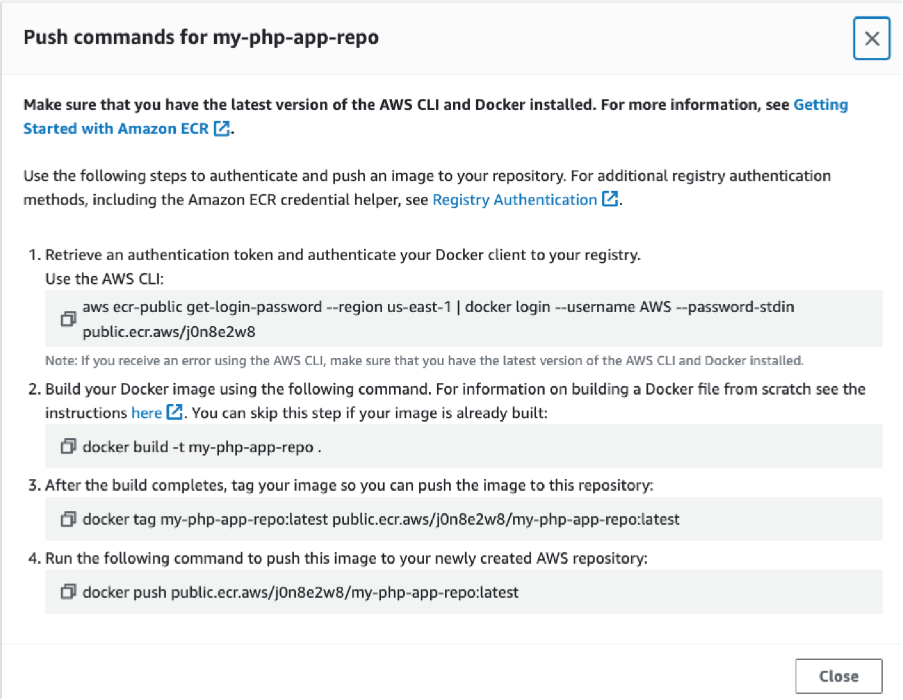
Repositories (1)

	Repository name ▲	URI	Created at
○	my-php-app-repo	public.ecr.aws/j0n8e2w8/my-php-app-repo	March 08, 2024, 09:35:41 (UTC+03)

3. Upload/Push Docker Image to ECR Repository (Using AWS CLI):

Note: This step requires the AWS CLI installed and configured with your IAM user credentials (access key and secret key).

- Follow the commands provided



Push commands for my-php-app-repo

Make sure that you have the latest version of the AWS CLI and Docker installed. For more information, see Getting Started with Amazon ECR.

Use the following steps to authenticate and push an image to your repository. For additional registry authentication methods, including the Amazon ECR credential helper, see Registry Authentication.

1. Retrieve an authentication token and authenticate your Docker client to your registry.
   Use the AWS CLI:

```
aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --password-stdin
public.ecr.aws/j0n8e2w8
```

Note: If you receive an error using the AWS CLI, make sure that you have the latest version of the AWS CLI and Docker installed.

2. Build your Docker image using the following command. For information on building a Docker file from scratch see the instructions here. You can skip this step if your image is already built:

```
docker build -t my-php-app-repo .
```

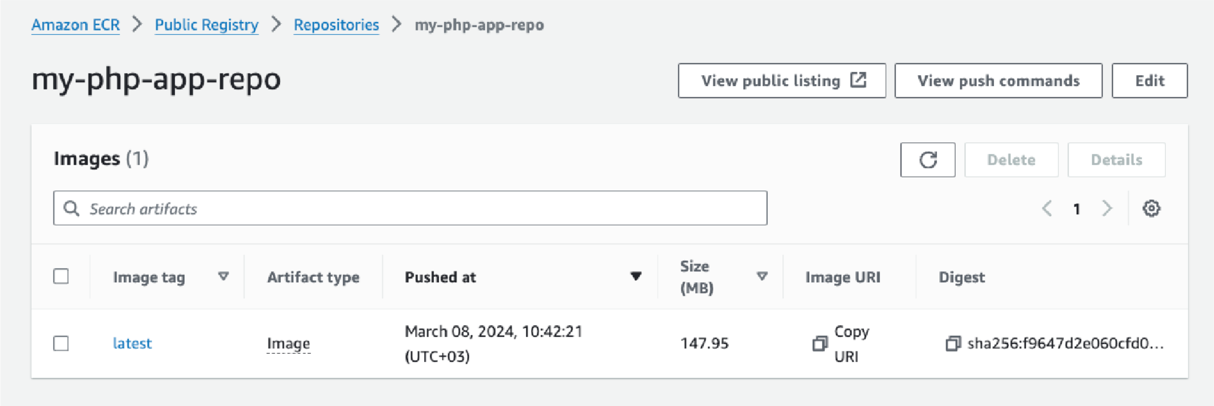
3. After the build completes, tag your image so you can push the image to this repository:

```
docker tag my-php-app-repo:latest public.ecr.aws/j0n8e2w8/my-php-app-repo:latest
```

4. Run the following command to push this image to your newly created AWS repository:

```
docker push public.ecr.aws/j0n8e2w8/my-php-app-repo:latest
```

- Once it works you should be able to view the image in your repository



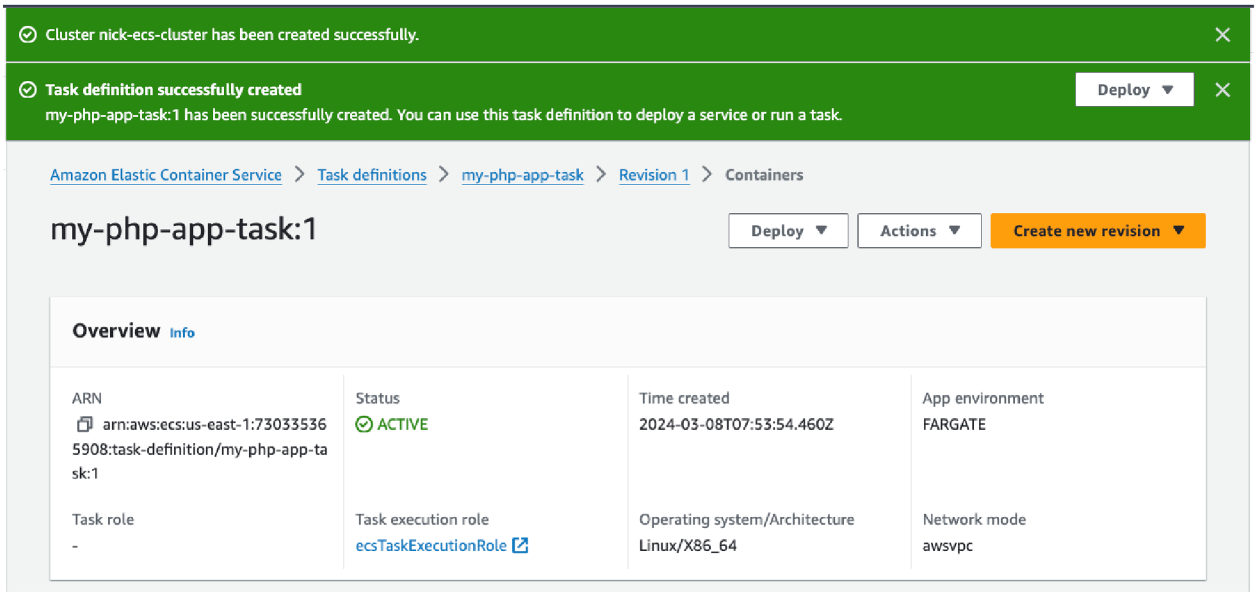
my-php-app-repo

Images (1)

□	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
□	latest	Image	March 08, 2024, 10:42:21 (UTC+03)	147.95	Copy URI	sha256:f9647d2e060cfd0...

4. Create an ECS Task Definition with Fargate Launch Type:

1. In the ECS Management Console, navigate to "Task Definitions".
2. Click on "Create task definition".
3. Choose a compatibility option (Fargate)


Deploy
Amazon Elastic Container Service > Task definitions > my-php-app-task > Revision 1 > Containers
Deploy
Actions
Create new revision
Info

Cluster nick-ecs-cluster has been created successfully.

Task definition successfully created
my-php-app-task:1 has been successfully created. You can use this task definition to deploy a service or run a task.

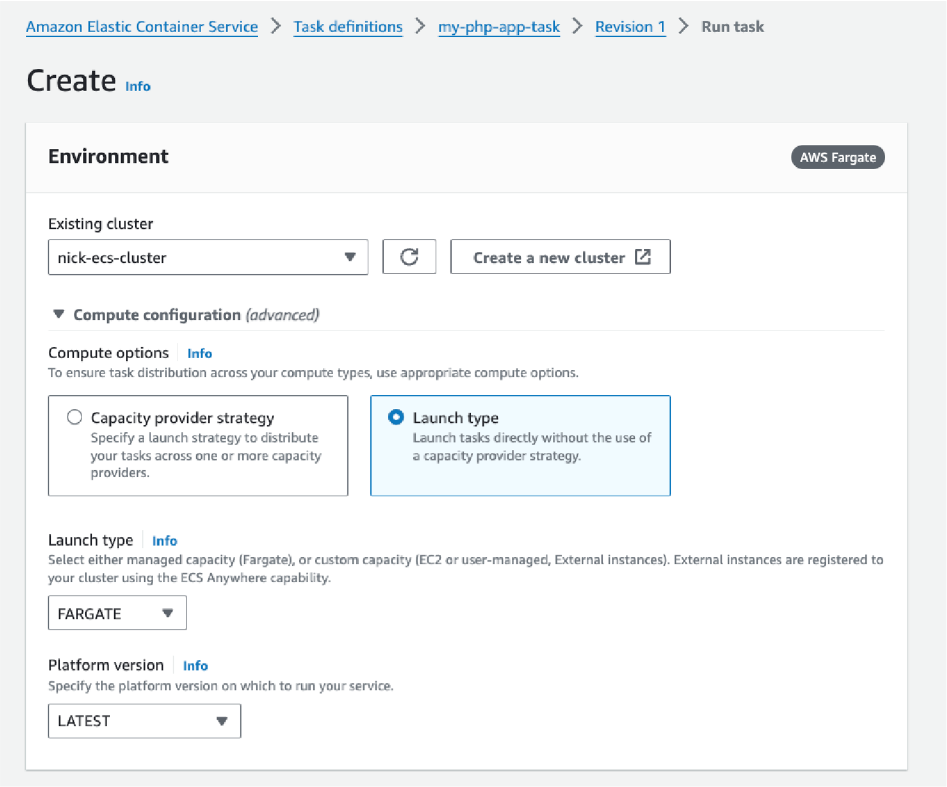
my-php-app-task:1

Overview

ARN	Status	Time created	App environment
arn:aws:ecs:us-east-1:730335365908:task-definition/my-php-app-task:1	ACTIVE	2024-03-08T07:53:54.460Z	FARGATE
Task role	**Task execution role**	**Operating system/Architecture**	**Network mode**
-	ecsTaskExecutionRole	Linux/X86_64	awsvpc

5. Launch a New Fargate Task:

1. In the ECS Management Console, navigate to "Tasks".



Amazon Elastic Container Service > Task definitions > my-php-app-task > Revision 1 > Run task

Create Info

Environment — AWS Fargate

Existing cluster: nick-ecs-cluster

▼ Compute configuration *(advanced)*

Compute options Info
To ensure task distribution across your compute types, use appropriate compute options.

○ Capacity provider strategy
Specify a launch strategy to distribute your tasks across one or more capacity providers.

● Launch type
Launch tasks directly without the use of a capacity provider strategy.

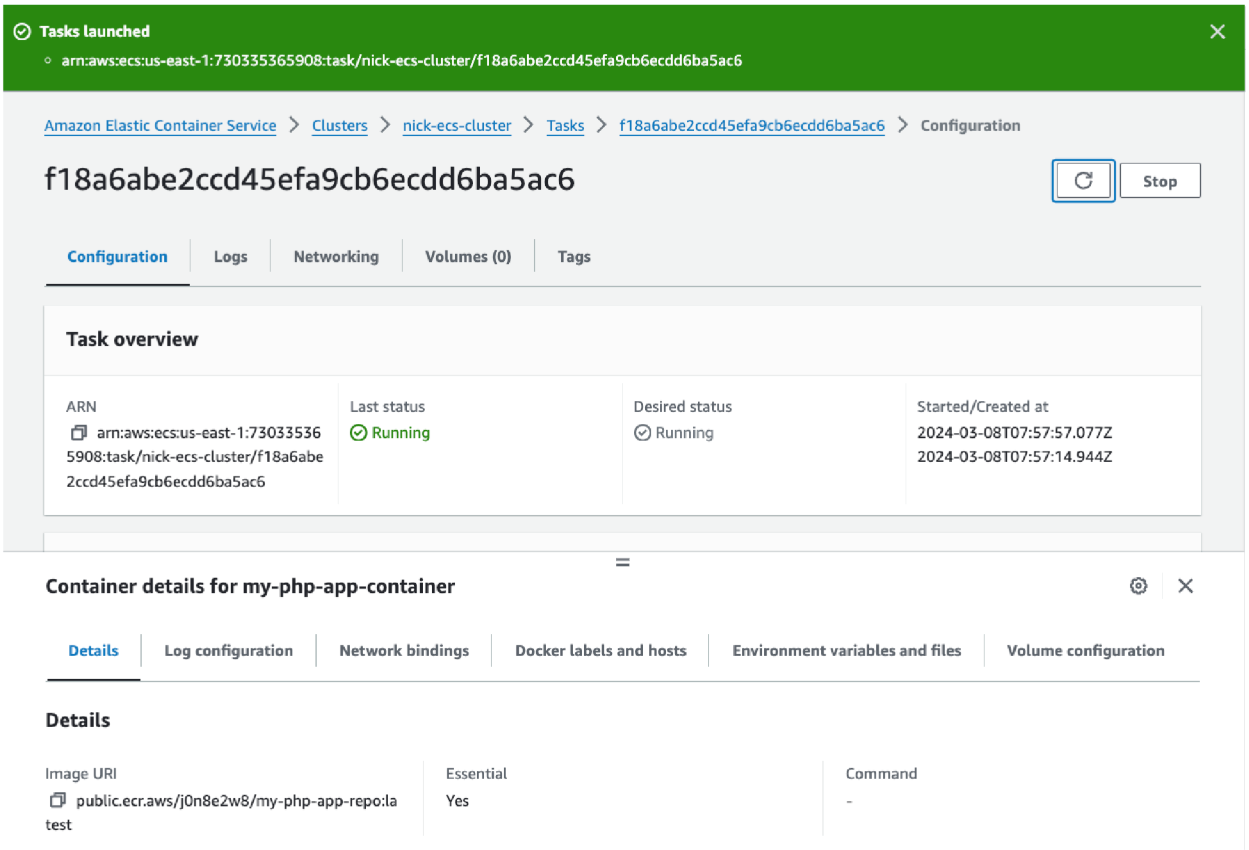
Launch type Info
Select either managed capacity (Fargate), or custom capacity (EC2 or user-managed, External instances). External instances are registered to your cluster using the ECS Anywhere capability.

FARGATE

Platform version Info
Specify the platform version on which to run your service.

LATEST

2. Click on "Run new task".
3. Choose your cluster and task definition.
4. Under "Launch type", select "Fargate".
5. Configure the number of tasks to launch (usually 1).
6. (Optional) Configure advanced launch settings like resource allocation for Fargate tasks.
7. Click on "Run" to launch the task using your created task definition.



Tasks launched
- arn:aws:ecs:us-east-1:730335365908:task/nick-ecs-cluster/f18a6abe2ccd45efa9cb6ecdd6ba5ac6

Amazon Elastic Container Service > Clusters > nick-ecs-cluster > Tasks > f18a6abe2ccd45efa9cb6ecdd6ba5ac6 > Configuration

f18a6abe2ccd45efa9cb6ecdd6ba5ac6

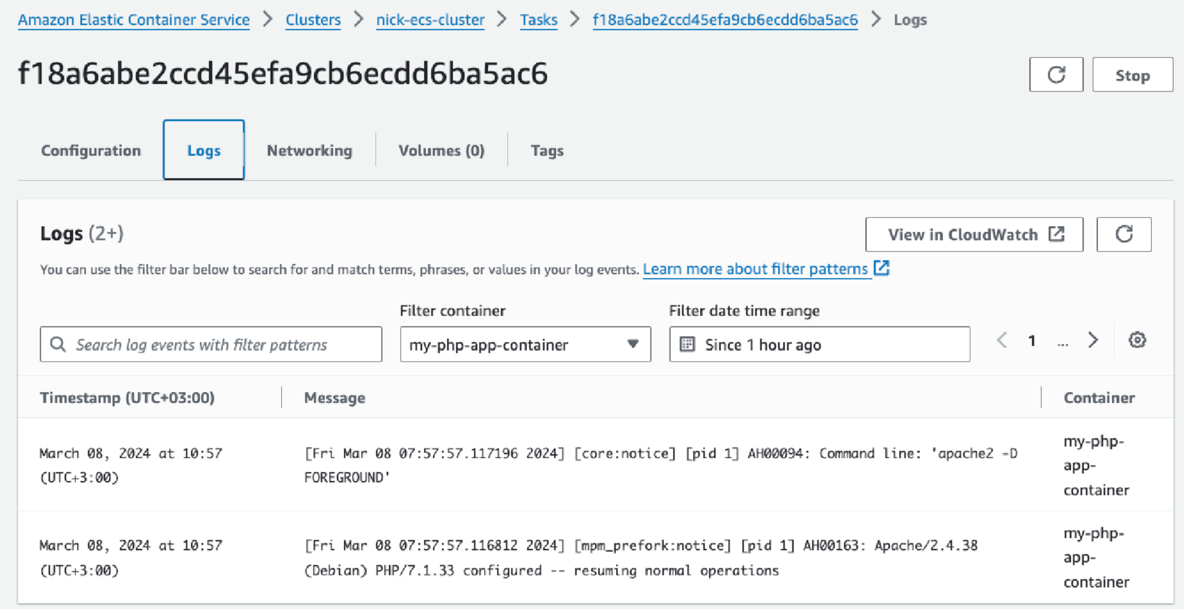
Task overview

ARN	Last status	Desired status	Started/Created at
arn:aws:ecs:us-east-1:730335365908:task/nick-ecs-cluster/f18a6abe2ccd45efa9cb6ecdd6ba5ac6	✓ Running	✓ Running	2024-03-08T07:57:57.077Z 2024-03-08T07:57:14.944Z

Container details for my-php-app-container

Details

Image URI	Essential	Command
public.ecr.aws/j0n8e2w8/my-php-app-repo:latest	Yes	-


Amazon Elastic Container Service > Clusters > nick-ecs-cluster > Tasks > f18a6abe2ccd45efa9cb6ecdd6ba5ac6 > Logs
f18a6abe2ccd45efa9cb6ecdd6ba5ac6
Stop
Configuration
Logs
Networking
Volumes (0)
Tags
Logs (2+)
View in CloudWatch

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

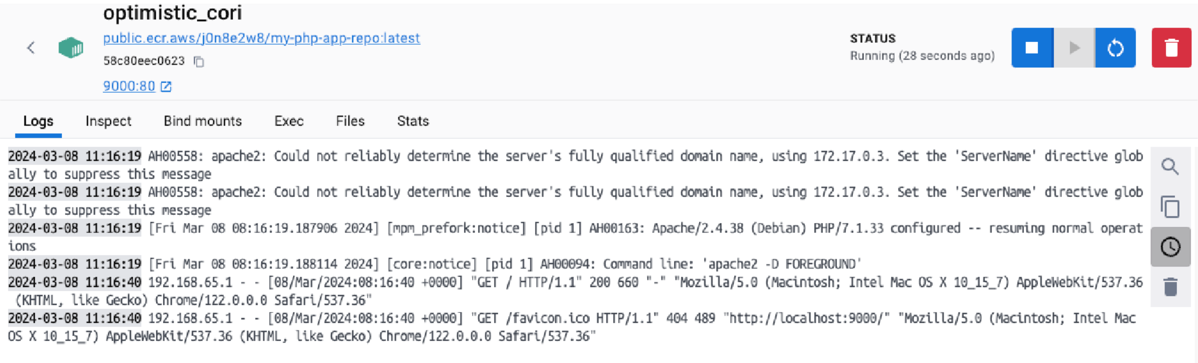
Filter container: my-php-app-container

Filter date time range: Since 1 hour ago

Timestamp (UTC+03:00)	Message	Container
March 08, 2024 at 10:57 (UTC+3:00)	[Fri Mar 08 07:57:57.117196 2024] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'	my-php-app-container
March 08, 2024 at 10:57 (UTC+3:00)	[Fri Mar 08 07:57:57.116812 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.1.33 configured -- resuming normal operations	my-php-app-container

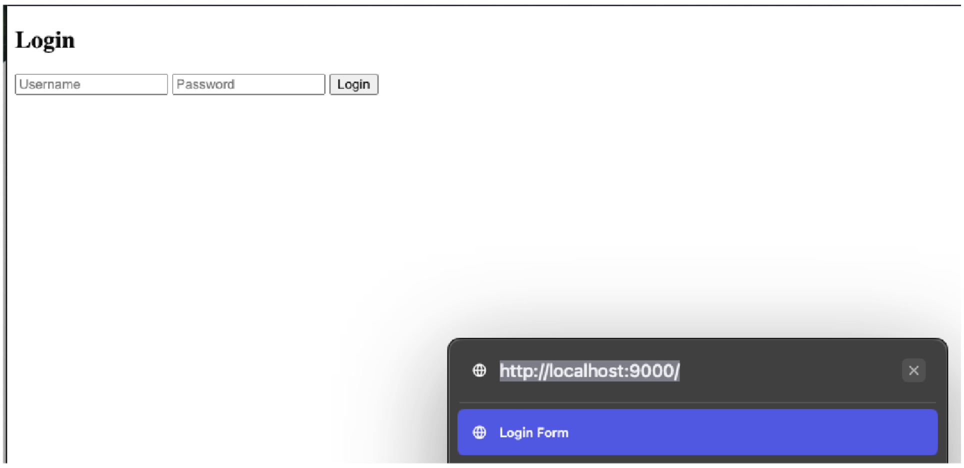
Tested it locally

```
docker run -p 9000:80 public.ecr.aws/j0n8e2w8/my-php-app-repo:latest
```

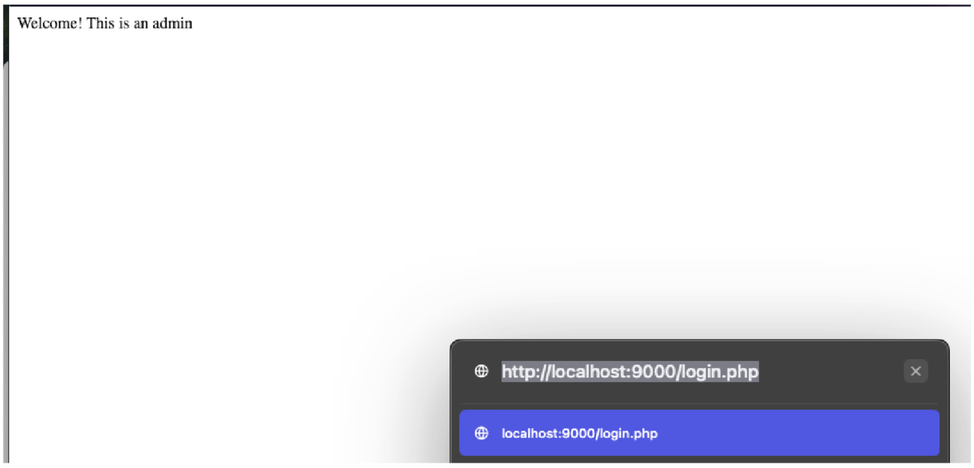


```
2024-03-08 11:16:19 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive glob
ally to suppress this message
2024-03-08 11:16:19 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive glob
ally to suppress this message
2024-03-08 11:16:19 [Fri Mar 08 08:16:19.187906 2024] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.38 (Debian) PHP/7.1.33 configured -- resuming normal operat
ions
2024-03-08 11:16:19 [Fri Mar 08 08:16:19.188114 2024] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
2024-03-08 11:16:40 192.168.65.1 - - [08/Mar/2024:08:16:40 +0000] "GET / HTTP/1.1" 200 660 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36"
2024-03-08 11:16:40 192.168.65.1 - - [08/Mar/2024:08:16:40 +0000] "GET /favicon.ico HTTP/1.1" 404 489 "http://localhost:9000/" "Mozilla/5.0 (Macintosh; Intel Mac
OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36"
```

- Login page successful with task run on ECS on port 9000



- Login page successful with correct username & password.



Welcome! This is an admin

http://localhost:9000/login.php

localhost:9000/login.php