



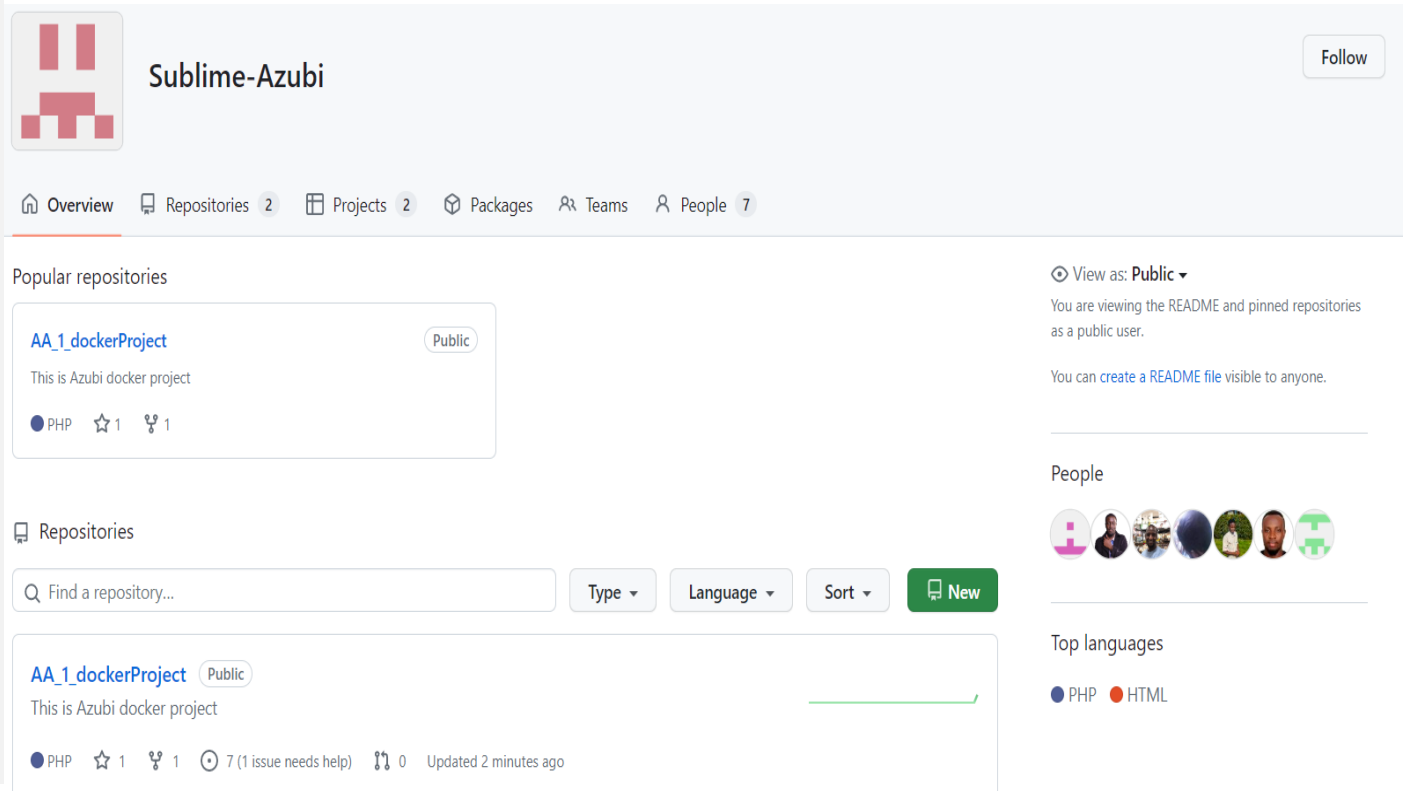
Azubi Africa

AWS SAA Project 1

TEAM SUBLIME PROJECT PRESENTATION

# Project Sublime Team members

- Amos Mogaka Nyaburi
- Brian Gitau
- Christopher Yongo
- Ebenezer Kuku
- Emmanuel Bassey
- Ifeanyi Okeibunor
- Eva Naomi Njoroge
- Nana Esi Ashun-Cobbina
- Uwimana Alphonsine



The screenshot shows the GitHub profile page for 'Sublime-Azubi'. The profile name is 'Sublime-Azubi' with a 'Follow' button. The navigation bar includes 'Overview', 'Repositories 2', 'Projects 2', 'Packages', 'Teams', and 'People 7'. The 'Popular repositories' section features a card for 'AA\_1\_dockerProject', which is a public repository in PHP with 1 star and 1 fork. The 'Repositories' section has a search bar and filters for 'Type', 'Language', and 'Sort', along with a 'New' button. Below the search bar, the 'AA\_1\_dockerProject' repository is listed again, showing it is public, in PHP, with 1 star, 1 fork, 7 issues (1 needs help), and 0 pull requests, updated 2 minutes ago. On the right side, there is a 'View as: Public' dropdown, a note about viewing the README and pinned repositories as a public user, a link to create a README file, a 'People' section with profile pictures, and a 'Top languages' section showing PHP and HTML.

Fork this repo at: [https://github.com/Sublime-Azubi/AA\\_1\\_dockerProject](https://github.com/Sublime-Azubi/AA_1_dockerProject)

# Task 1.

- ❑ Create a static HTML file.
- ❑ Create a Dockerfile.
- ❑ Build a docker image.
- ❑ Run the docker image in a container.
- ❑ Push the docker image to docker hub.



# Sample of html login script

```
<?php ?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/water.css@2/out/water.css">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>signup</title>
</head>
<body>
  <h1>Azubi Docker Log in </h1>
  <form name="login" action="signup-valid.php" method="post" autocomplete="off">
    <div>
      <label id="name" for="name" name="name">Username</label>
      <input type="text" name="name" placeholder="user name">
    </div>
    <div>
      <label for="password" name="password">password</label>
      <input id="password" type="password" name="password">
    </div>
    <div>
      <p><input type="submit" name="submit" value="Submit"></p>
    </div>
  </form>
</body>
</html>
```

Credits: <https://watercss.kognise.dev>  
For CSS styling.

## Azubi Docker Log in

Username

password

# Create a Dockerfile and Build image

Created Dockerfile in the root directory using variant that contains a web server `.php:<version>-cli - (php:8.0-apache)`

Docker build command

```
$ docker build -t php-apache .
```

Run the Docker image

```
$ docker run -it --rm --name m-apache-app php-apache
```

```
1 FROM php:8.0-apache
2
3 COPY . /var/www/html/
4 WORKDIR /var/www/html/
5
6 EXPOSE 80
7
```

# Push the image to Docker registry

Configure terminal with your docker id.

## **Command to push docker image**

```
$ docker image tag rhel-httpd:latest registry-host:5000/myadmin/rhel-httpd:latest
```

```
$ docker image push registry-host:5000/myadmin/rhel-httpd:latest
```

Pull the image using the below command on the terminal.

```
$ docker pull chrisjanabi/php-apache
```

## Task 2.

- ▶ Create a second HTML file with additional functionalities.
- ▶ Create a php file (server-side scripting) to process the inputs when we submit the form.
- ▶ Create a Dockerfile.
- ▶ Build a docker image & Run the docker image in a container.
- ▶ Push the docker image to docker hub and Amazon elastic container registry.
- ▶ Create Amazon ECS cluster. Task definition with Fargate launch type.





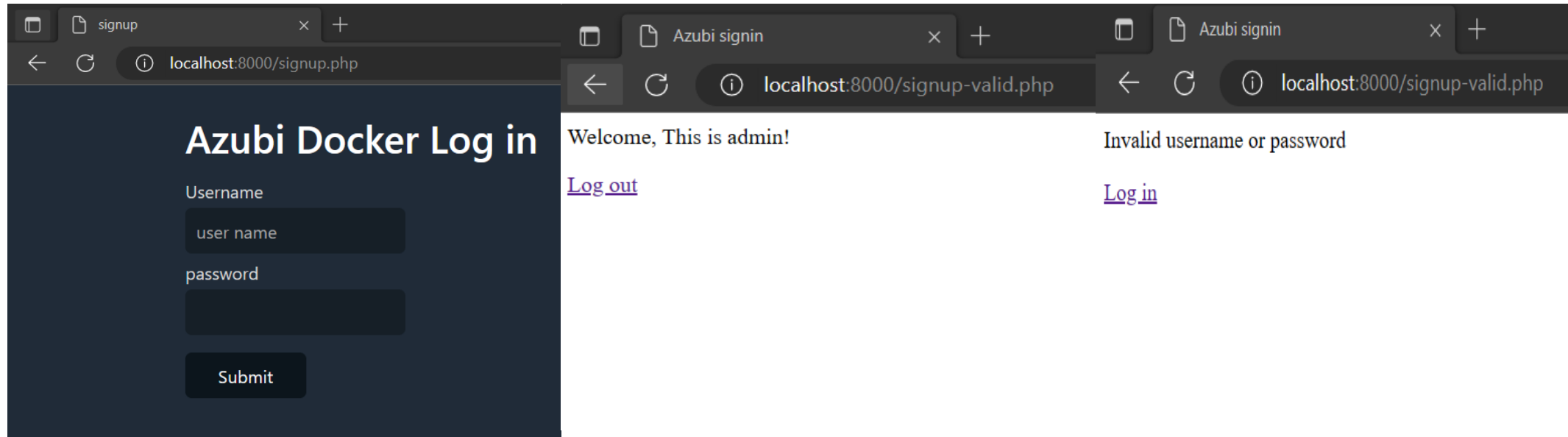
# Php login and form validation scripts

```
signup-valid.php
C: > Users > admin > Desktop > AA_1_dockerProject > signup-valid.php
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Azubi signin</title>
6   </head>
7   <body>
8
9     <?php
10      if(isset($_POST['submit'])){
11          $userName = $_POST['name'];
12          $password = $_POST['password'];
13          $folder_path = 'signup.php';
14
15          if ($userName !== 'johndoe' && $password !== '12345'){
16
17              print('Invalid username or password');
18              echo '<p><a href="' . $folder_path . '>Log in</a></p>';
19          }
20          }else{
21              print('Welcome, This is admin!');
22              echo '<p><a href="' . $folder_path . '>Log out</a></p>';
23          }
24      }
25  }
26 </body>
27 </html>
```

```
signup.php
C: > Users > admin > Desktop > AA_1_dockerProject > signup.php
1 <?php ?>
2
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6   <meta charset="UTF-8">
7   <meta http-equiv="X-UA-Compatible" content="IE=edge">
8   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/water.css@2/out
9   <meta name="viewport" content="width=device-width, initial-scale=1.0">
10  <title>signup</title>
11 </head>
12 <body>
13   <h1>Azubi Docker Log in </h1>
14   <form name="login" action="signup-valid.php" method="post" autocomplete="
15     <div>
16       <label id="name" for="name" name="name">Username</label>
17       <input type="text" name="name" placeholder="user name">
18     </div>
19     <div>
20       <label for="password" name="password">password</label>
21       <input id="password" type="password" name="password">
22     </div>
23     <div>
24       <p><input type="submit" name="submit" value="Submit"></p>
25     </div>
26   </form>
27 </body>
28 </html>
29
```



# Sample of php validation



Admin Username – johndoe    Admin Password – 12345

# Pushing Docker image to ECR

## Prerequisites for (ECS- Elastic Container Services) & (ECR- Elastic Container Registry)

- ▶ AWS CLI and docker desktop downloaded and installed.
- ▶ Configure CLI (aws cli configure – provide access ID and keys ). Ensure the user has programmatic access and able to execute the necessary tasks in ecr and ecs.
- ▶ On Amazon ecr console create image repository.
- ▶ Get the push commands from ecr repository console.

# Build, Tag and Push image to registry

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/v2j3g8s3
```

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 has already been built:



```
docker build -t php-apache .
```

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

```
docker tag php-apache:latest public.ecr.aws/v2j3g8s3/php-apache:latest
```

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

```
docker push public.ecr.aws/v2j3g8s3/php-apache:latest
```

Repository name	URI
<a href="#">php-apache</a>	 public.ecr.aws/v2j3g8s3/php-apache
<a href="#">sublime_task</a>	 public.ecr.aws/v2j3g8s3/sublime_task

# ECS with Fargate

In order to create and run ECS FARGATE task, you need to complete the following three steps:

- Have a docker repository (for ex. ECR) with an image that you want to run
- Create an ECS cluster and define a task with the image from ECR
- Run a task in the cluster

# Click on the ENI to get the Public IP

Search

Network interface ID = eni-089749758e3c4fab2

Clear filters

Name	Network interface ID	Subnet ID	VPC ID	Availability Zone	Security group n...
------	----------------------	-----------	--------	-------------------	---------------------

**Network interface: eni-089749758e3c4fab2**

IP addresses

Private IPv4 address 10.0.0.70	Private IPv4 DNS ip-10-0-0-70.ec2.internal	Elastic Fabric Adapter False
<b>Public IPv4 address 54.81.14.55</b>	Public IPv4 DNS ec2-54-81-14-55.compute-1.amazonaws.com	IPv6 addresses -
Secondary private IPv4 addresses -	Association ID -	Elastic IP address owner amazon
MAC address 0e:49:94:c7:92:eb	IPv4 Prefix Delegation -	IPv6 Prefix Delegation -

signup

Not secure 54.81.14.55/signup.php

Free Course - MAS... Free Course - MAS... Contribute to some...

## Azubi Docker Log in

Username

password

# Project Management Tool

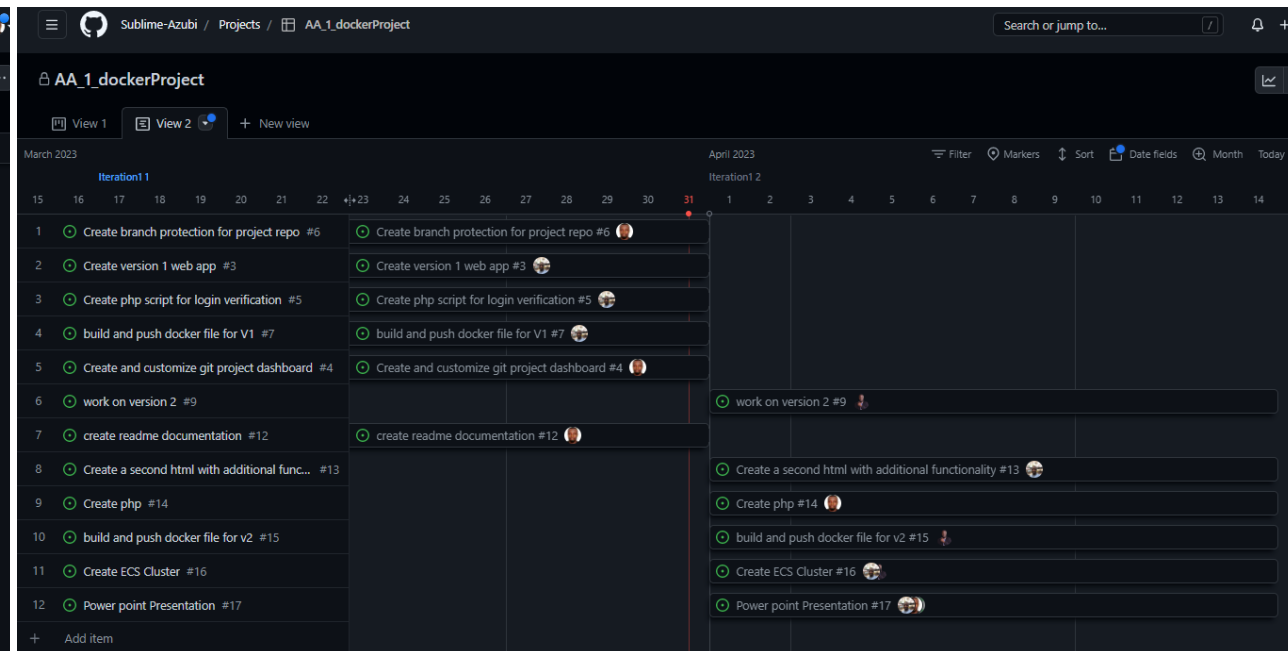
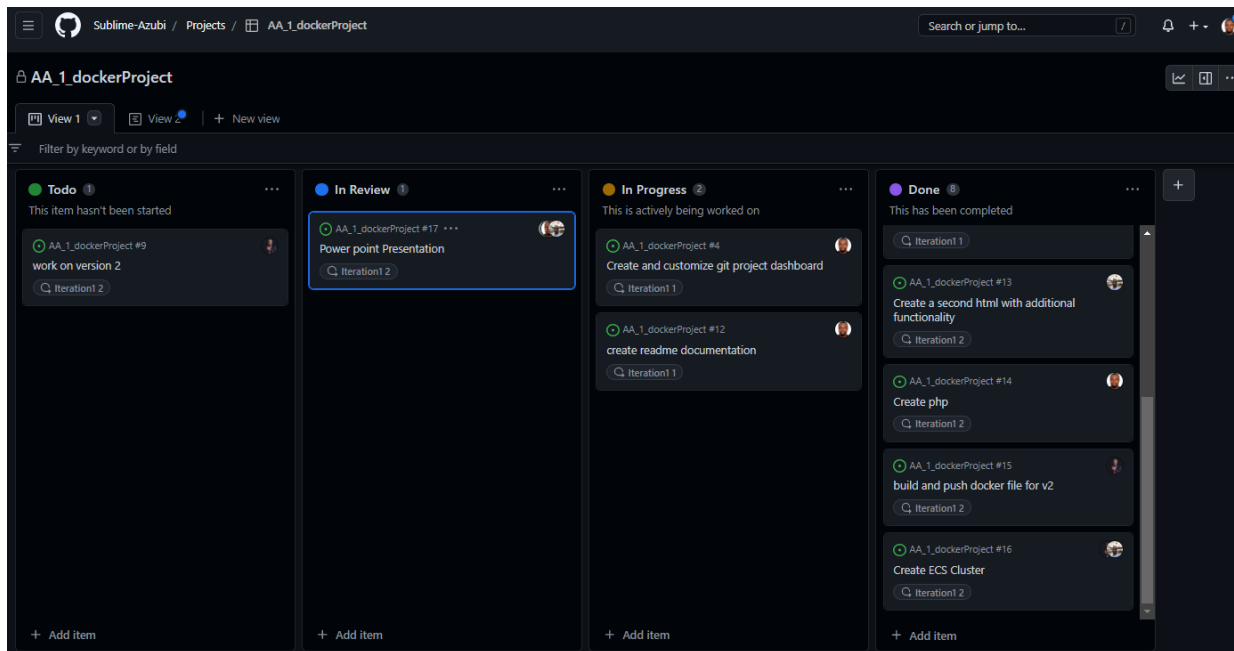
## GITHUB PROJECT

- ▶ Github Project was chosen as best fit for our project
  - ▶ It helps us to track our progress on each task on our project
  - ▶ It is open source and free.
  - ▶ As part of github tools, it allows us to link our project dashboard to github “push” and “commit” operations

# GITHUB PROJECT DASHBOARD

Left side image shows our github project tasks in a customizable dashboard

Right side image shows our github project tasks in in different iterations and defined in task1 and task 2





# Resources

- ▶ Docker Website: [https://hub.docker.com/\\_/php](https://hub.docker.com/_/php)
- ▶ Water.CSS: <https://watercss.kognise.dev>
- ▶ Github: <https://docs.github.com/en>
- ▶ Youtube: <https://www.youtube.com/>
- ▶ Whizlabs: <https://business.whizlabs.com>
- ▶ AWS: <https://docs.aws.amazon.com>



THANK YOU.