**Lab 10: Advanced Security with Docker and AWS**

**Objective:**

Implement best practices for securing Docker containers on AWS.

**Tasks:**

1. Configure Docker to use TLS for secure remote access.

2. Implement AWS Identity and Access Management (IAM) for ECS tasks.

**Documentation:**

1. Security considerations with Docker.

2. Leveraging AWS security features for Docker deployments.

**Tasks:**

**1. Configure Docker to use TLS for secure remote access.**

**Why?** By default, Docker's daemon listens on a UNIX socket. If you want to access the daemon remotely, you need to secure it using TLS to ensure encrypted communication.

**Steps:**

Generate CA, server, and client keys.

| mkdir -pv ~/certs cd ~/certs openssl genrsa -out ca-key.pem 4096 openssl req -new -x509 -days 365 -key ca-key.pem -sha256 -out ca.pem openssl genrsa -out server-key.pem 4096 openssl req -subj "/CN=$HOST" -sha256 -new -key server-key.pem -out server.csr |
| --- |

Sign the public key with our CA.

| openssl x509 -req -days 365 -sha256 -in server.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out server-cert.pem |
| --- |

Create client keys.

| openssl genrsa -out client-key.pem 4096 openssl req -subj '/CN=client' -new -key client-key.pem -out client.csr openssl x509 -req -days 365 -sha256 -in client.csr -CA ca.pem -CAkey ca-key.pem -CAcreateserial -out client-cert.pem |
| --- |

Start Docker with TLS.

| dockerd --tlsverify --tlscacert=ca.pem --tlscert=server-cert.pem --tlskey=server-key.pem -H=0.0.0.0:2376 |
| --- |

**2. Implement AWS Identity and Access Management (IAM) for ECS tasks.**

**Why?** AWS IAM allows you to manage access to AWS services and resources securely. With IAM, you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources.

**Step 2.1:** Create a new IAM Role for ECS tasks:

In the AWS Management Console, navigate to **IAM > Roles > Create Role.** Choose ECS as the service and ECS Task as the use case.

**Step 2.2: Attach permissions to the role:**

Attach relevant permissions to the role based on what your Docker containers need to access. For instance, if your container needs to read from an S3 bucket, attach the '**AmazonS3ReadOnlyAccess**' policy.

**Step 2.3: Assign the IAM Role to your ECS Task Definition:**

When creating or updating an ECS Task Definition in the AWS Management Console, under the task execution IAM role, select the role you just created.

**Benefits of this Lab:**

**Enhanced Security:** By implementing TLS for Docker and leveraging AWS IAM, you ensure that your Docker containers are securely deployed and accessed.

**Best Practices:** This lab teaches students the industry's best practices for securing Docker on AWS.

**Practical Experience:** Students gain hands-on experience with Docker and AWS, two of the most widely used technologies in the industry.

**Troubleshooting Tips:**

**1**. Docker TLS Issues: Ensure all the paths in the daemon.json file are correct and the necessary certificates are in place.

**2.** IAM Permissions: If your Docker containers are having issues accessing AWS resources, double-check the permissions attached to the IAM role. Ensure that the ECS task is using the correct role.

**3.** ECS Tasks: For ECS task-related issues, check the AWS CloudWatch logs for clues. Ensure that your task definition is correctly set up.