Report:

➔ **Containerization vs Serverless**

	Container	**Serverless**
Deployment	User has full control over the environment and consistency across different platforms	Limited control over the environment as compared to containers
	More setup and configuration will be required for scaling, networking, and security	Automatic scaling offered by the services
Cost	Suitable for long running applications as the resources are allocated accordingly	It generally follows a pay-per-execution model where user pays for the execution of the function/application. A long running application would be expensive
	Need to pay even during downtime, which might increase cost when traffic is low	Pay per execution model hence no cost during downtime
Performance	No cold starts	Can suffer from cold start delays when application has been idle for long time
	Containers are always running making it idle for continuous workflows	Offers fast execution for burst workloads (short and quick)
	Performance is dependent on the underlying resources	Performance can be optimized (as used in the assignment by increasing memory and time out)
	Containers have higher limits for the memory usage than serverless	Lambda had maximum memory usage-15GB

Conclusion for the Image Classification Application:

- **AWS Lambda** was much faster in execution with it taking 33 seconds to complete the training process and give an accuracy Score
- **Docker** on local host took a long time perhaps due to the laptop's configuration.
- Since it is a short sporadic task, AWS Lambda is the preferred choice for our image classification application. It is faster, pay-per-execution, and can be scaled automatically.