

Deploying Everything with Amazon ECS

Pushing Application Changes without Downtime

Build a new version of the application

```
docker-compose build
```

Tag the Dockerzon image

```
docker tag -f dockerzon_dockerzon:latest \  
xxx.dkr.ecr.us-east-1.amazonaws.com/dockerzon/dockerzon:latest
```

- Replace my AccountId (xxx) with yours

Authenticate your Docker client

```
aws ecr get-login
```

Push the Dockerzon image to your repository

```
docker push xxx.dkr.ecr.us-east-1.amazonaws.com/dockerzon/dockerzon
```

- Replace my AccountId (xxx) with yours

View the web application in your browser

```
http://the_dns_name_of_your_load_balancer
```

- Replace the_dns_name_of_your_load_balancer with your ELB's DNSName

Register a new web task definition

```
# Make sure you're in the production/ folder when executing this command  
aws ecs register-task-definition \  
--cli-input-json file://web-task-definition.json
```

Update the web service

```
aws ecs update-service --cluster production --service web \  
--task-definition web --desired-count 2
```

Register a new worker task definition

```
# Make sure you're in the production/ folder when executing this command  
aws ecs register-task-definition \  
--cli-input-json file://worker-task-definition.json
```

Update the worker service

```
aws ecs update-service --cluster production --service worker \  
--task-definition worker --desired-count 1
```