



Deploy an App Across Accounts



mallangiharinathreddy727@gmail.com

Hellooooo from HARINATH at Nextwork!

If you can see this, you've deployed my app... nice work!

You've unlocked my secret code: BIRYANI

Something I've learnt about you today is...

And here's a special image chosen by me:

Custom Image

A rectangular placeholder box with a thin gray border, labeled "Custom Image" at the top left corner.

Introducing Today's Project!

In this special multiplayer project, I'm working with a buddy on cross-account deployment! What this means is my buddy and I will be creating our individual applications, containerize them, use ECR to swap our containerized apps and try deploy it!

What is Amazon ECR?

Amazon ECR is an AWS service that lets us store our container images in the cloud. In more technical terms, it is a managed container registry.

One thing I didn't expect...

One thing I didn't expect in this project was being able to run Natasha's first ECR push command and seeing "login succeeded". It was a learning to know that authenticated us to the ECR the service not the images and the resources inside.

This project took me...

This project took me 3 Hours to complete include documentation.

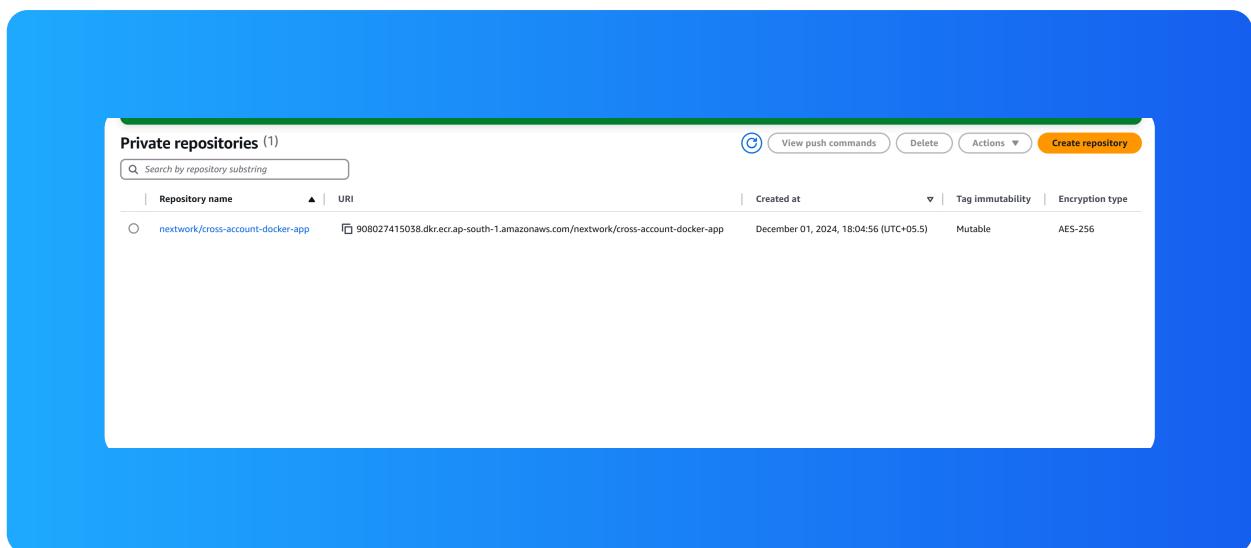
Creating a Docker Image

I set up a Dockerfile and an index.html. Both files are needed- the Dockerfile tells Docker what kind of container image I want to build, while the index.html file is required to customise my webpage.

My Dockerfile tells Docker to start my custom container image as a copy of an existing container image called Nginx, and then add our modification to it i.e. use our own index.html file instead of the Nginx default.

I also set up an ECR repository

ECR stands for Elastic Container Registry. It is important because it easy for you to store, manage, and deploy your container images.



Set Up AWS CLI Access

AWS CLI can let me run ECR commands

AWS CLI is a way to access the AWS services in our terminal. The CLI asked for my credentials because it doesn't know who we are and wants the right access key to log us into the right account.

'To enable CLI access, I set up a new IAM user with the permission policy "AmazonEC2ContainerRegistryFullAccess". I also setup an Access key for this user, which means I can login in my terminal with my new IAM user.

To pass my credentials to the AWS CLI, I ran the command aws-configure and I had to provide my access key id, secret access key, and the region which I set up my ECR.

```
Unable to locate credentials. You can configure credentials by running "aws configure".
Error: Cannot perform an interactive login from a non TTY device
C:\Users\mallu\Desktop\DockerECR>aws configure
AWS Access Key ID [None]:
```

Pushing My Image to ECR

Push commands are terminal commands that let us push our resources into the cloud. In this context, we are using Amazon ECR which is giving us push commands to push our container images from our local computer into the cloud repositories in ECR.

There are three main push commands

To authenticate Docker with my ECR repo, I used the command "aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 908027415038.dkr.ecr.ap-south-1.amazonaws.com" to get access to ECR using my Credentials.

To push my container image, I ran the command docker push. Pushing means we have uploaded our container image to the cloud (ECR repository).

When I built my image, I tagged it with the label 'latest' This means latest is a commonly used tag to represent the most recent version. It always pulls the most up-to-date version.

Resolving Permission Issues

When I tried pulling my project buddy's container image for the first time, I saw the error 403 forbidden. This was because the repository we created with ECR were private, so we need to give each other specific permissions to gain access.

To resolve each other's permission errors, my buddy and I updated our ECR repository's permission settings . Now we have specifically allowed each other's ECR-Access user to have the permission to download container images in our repository.

```
C:\Users\mall\Desktop\DockerECR>docker pull 306662062493.dkr.ecr.us-west-2.amazonaws.com/nextwork/cross-account-docker-app:latest
Error response from daemon: pull access denied for 306662062493.dkr.ecr.us-west-2.amazonaws.com/nextwork/cross-account-docker-app, repository does not exist or may require 'docker login': denied: User: arn:aws:iam::908027415038: user/ECR-Access is not authorized to perform: ecr:BatchGetImage on resource: arn:aws:ecr:us-west-2:306662062493:repository/nextwork/cross-account-docker-app because no resource-based policy allows the ecr:BatchGetImage action
```



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

