Deploy an App Across Accounts

DI dimashamadhushani413@gmail.com

Hellloooo from Dimasha at Nextwork!

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

You've unlocked my secret code: Sea Food

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

And here's a special image chosen by me:

Introducing Today's Project!

In this special multiplayer project, I'm working with a buddy to build an app with a fun twist! We have the same difficulty levels, but each app has a unique tweak. Teamwork is key sharing ideas and learning together will make this great.

What is Amazon ECR?

Amazon ECR (Elastic Container Registry) is a managed service for storing Docker images. In today's project, you authenticated Docker with ECR, pushed an image to the registry, then pulled and ran it locally for seamless management.

One thing I didn't expect...

One thing I didn't expect in this project was the complexity of handling cross-architecture compatibility when pushing and pulling container images between different systems (eg., AMD64 and ARM64). It added an extra layer of consideration in process.

This project took me...

This project took me around 3-4 hours to complete. It involved setting up AWS ECR, building and pushing the container images, handling architecture compatibility issues, and troubleshooting along the way.

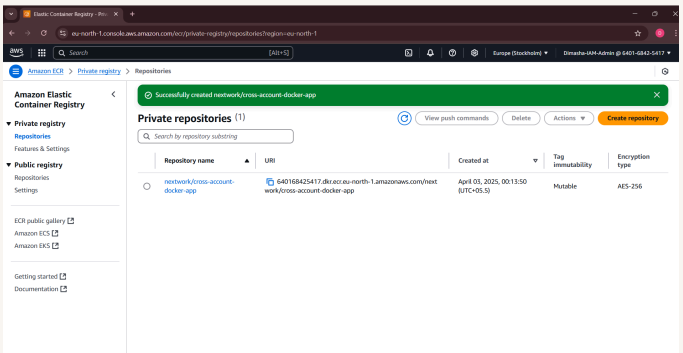
Creating a Docker Image

I set up a Dockerfile and an index.html file. Both files are needed because the Dockerfile defines the instructions to build a Docker image, while index.html file contains the content that will be served by the nginx web server inside the container.

My Dockerfile tells Docker to use the nginx:latest image as the base, and then copy the index.html file into the container's HTML directory. This sets up the nginx web server to serve the content from index.html when the container is run.

I also set up an ECR repository

ECR stands for Elastic Container Registry. It is important because it securely stores, manages, and deploys container images, ensuring teams always have access to the latest version without version conflicts.



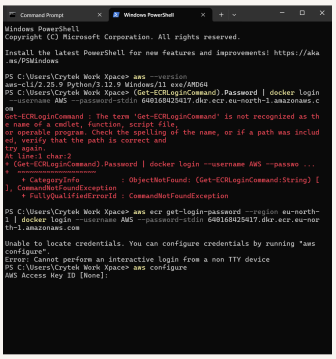
Set Up AWS CLI Access

AWS CLI can let me run ECR commands

AWS CLI is a tool for managing AWS services from the terminal. It asked for my credentials to authenticate access to AWS resources, like pushing images to ECR, managing S3 storage, or deploying apps securely.

To enable CLI access, I set up a new IAM user with the AmazonEC2ContainerRegistryFullAccess permission. I also set up an access key for this user, which means I can authenticate securely in the terminal to push container images to Amazon ECR.

To pass my credentials to AWS CLI, I ran aws configure. I provided the Access Key ID, Secret Access Key, and AWS region code for my ECR repository. I left the output format blank, using the default JSON format.



Pushing My Image to ECR

Push commands are a set of terminal commands provided by Amazon ECR that help you upload (or "push") your local Docker image to a specific repository in the cloud. They include authentication, tagging, and the push itself.

There are three main push commands

To authenticate Docker with my ECR repo,I used command"aws ecr get-login-password --region <your-region> | docker login --username AWS --password-stdin <your-account-id>.dkr.ecr.<your region>.amazonaws.com"This command Lets Docker access ECR securely

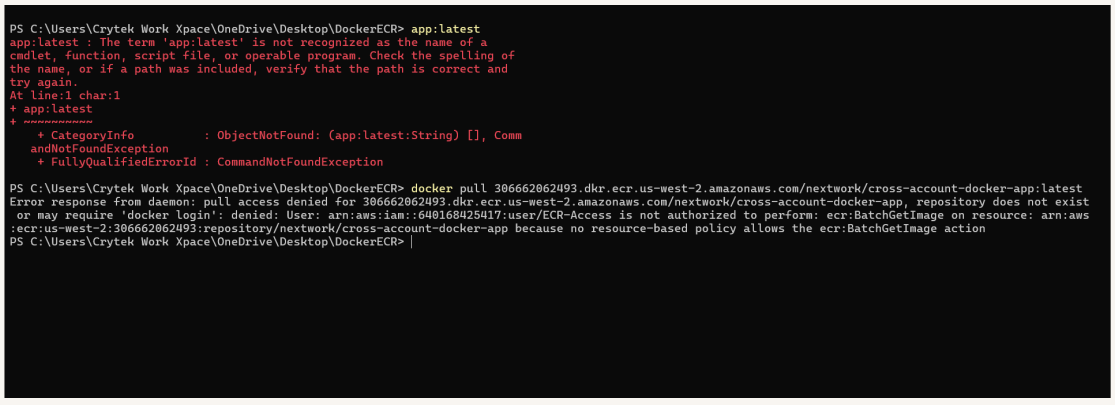
To push my container image, I ran the command: "docker push 640168425417.dkr.ecr.eu-north-1.amazonaws.com/nextwork/cross-account-docker-app:latest" Pushing means uploading my image to ECR so others can pull and use it.

When I built my image, I tagged it with the label latest. This means the image is marked as the most recent version. Tagging it as latest helps identify and pull the most up-to-date build easily during deployments or when sharing the image.

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 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've specifically allowed each other's ECR-Access user to have the permission to download container images in our ECR repository.



NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for more projects