Dockerizing a React Vite application and moving it into AWS ECS

Docker is a tool that allows developers to package and deploy applications in a lightweight, portable container. This makes it easy to run the same application on different environments and makes it easier to scale and manage applications in the cloud. In this blog post, we will look at how to dockerize a React Vite application and move it into AWS ECS (Amazon Elastic Container Service).

Before we start, it is important to have the following prerequisites:

- Node.js and npm installed on your machine
- An AWS account and access to the AWS Management Console
- The AWS CLI (Command Line Interface) installed on your machine
- Docker installed on your machine

Step 1: Setting up the React Vite application

If you don't already have a React Vite application, you can create one by running the following

command: ***npx create-vite-app my-app***

This will create a new directory called my-app with a basic React Vite application inside. Navigate to the my-app directory and start the development server by running the following command:

cd my-app

npm run dev

The development server should start and you should be able to see the application running on http://localhost:3000.

Step 2: Dockerizing the React Vite application

To dockerize the React Vite application, we will need to create a Dockerfile that specifies the instructions for building the Docker image. In the root directory of the application, create a file called Dockerfile and add the following content:

FROM node:14-alpine

WORKDIR /app

COPY package.json package-lock.json ./

RUN npm ci COPY . .

EXPOSE 3000

CMD ["npm", "run", "start"]

This Dockerfile specifies that we are using the node:14-alpine base image, which is a lightweight version of Node.js. It then sets the working directory to /app and copies the package.json and package-lock.json files into the working directory. It then runs the npm ci command to install the dependencies specified in the package.json file. Finally, it copies the rest of the application code into the working directory and exposes the default React Vite port of 3000.

To build the Docker image, run the following command: docker build -t my-app .

This will build a Docker image with the name my-app. You can verify that the image has been created by running the docker images command.

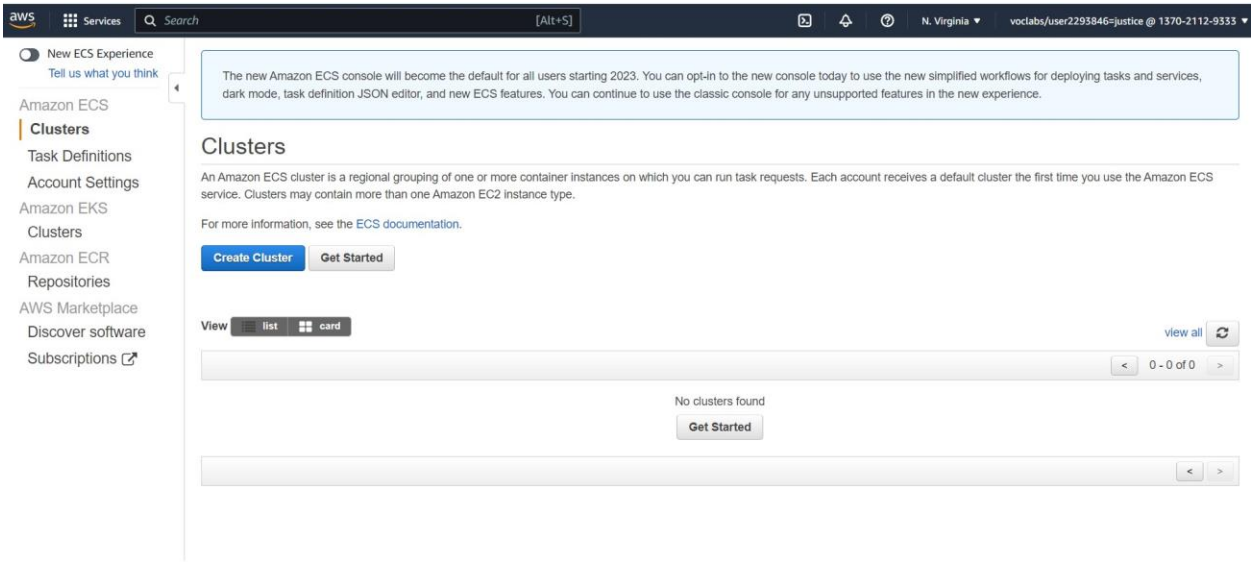
To run the Docker image, use the following command: docker run -p 3000:3000 my-app

This will start the Docker container and you should be able to see the React Vite application running on http://localhost:3000.

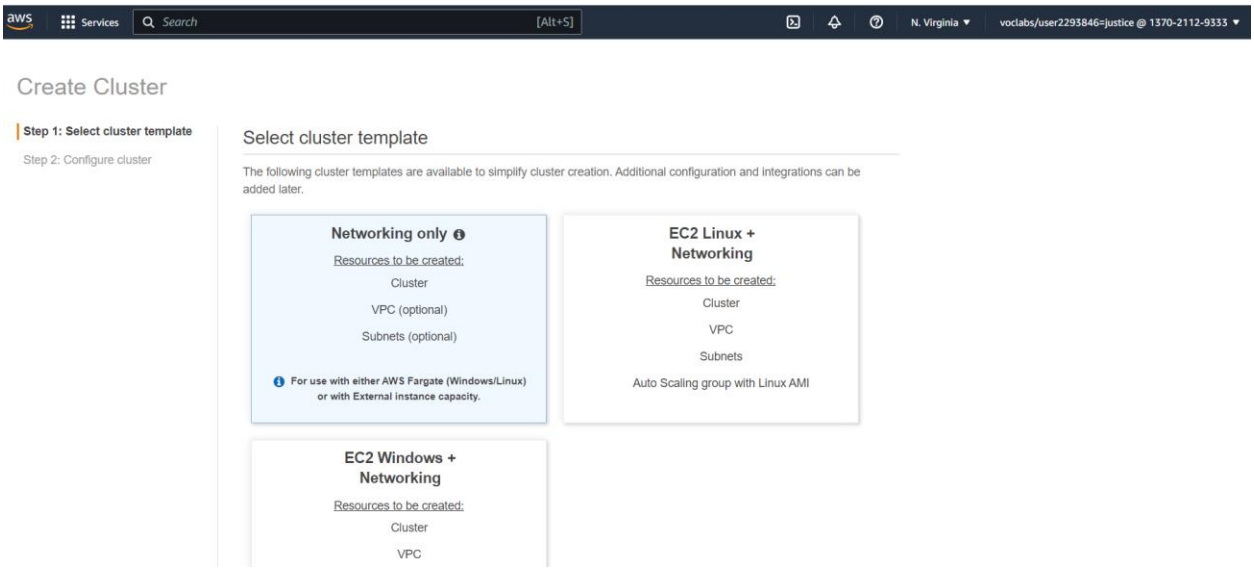
Step 3: Moving the Dockerized React Vite application to AWS ECS

To move the Dockerized React Vite application to AWS ECS, we will need to create an ECS cluster and deploy the Docker image to it.

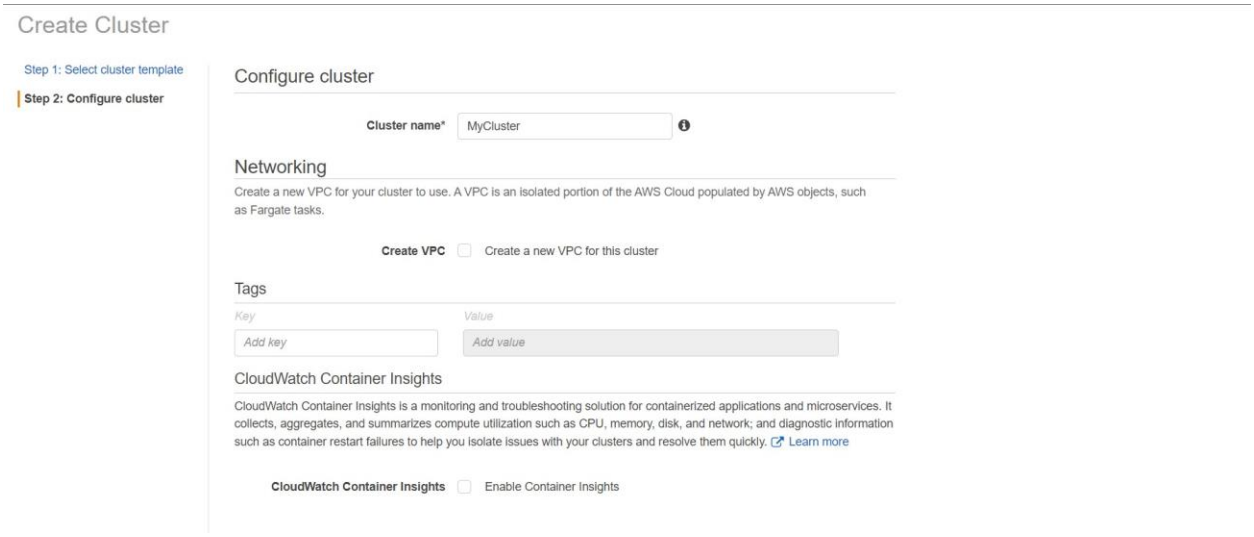
1. Log in to the AWS Management Console and navigate to the ECS dashboard.
2. Click on the "Create cluster" button.



3. Select the "Networking only" cluster type



4. Name your cluster and choose a region, then click on the "Create" button.



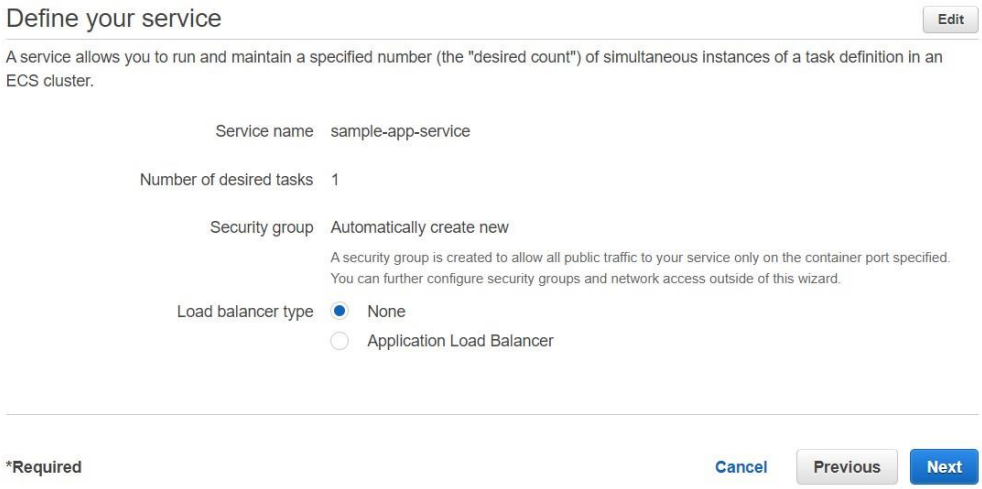
5. Once the cluster has been created, click on the "Create service" button.

6. Select the "EC2" launch type and choose the cluster that you just created.

7. Select the Docker image that you built earlier as the task image and specify the port mapping.

8. Configure the desired number of tasks and click on the "Next step" button.

9. Review your service settings and click on the "Create service" button.



10. Wait for the service to be created and then click on the "View service" button to see the details of the service.

11. Click on the "Tasks" tab and you should see the tasks running in the cluster.

12. To access the application, you can either use the public IP address of the EC2 instance or you can use a load balancer to distribute traffic to the tasks.

That's it! You have successfully dockerized your React Vite application and deployed it to AWS ECS. With this setup, you can easily scale and manage your application in the cloud.