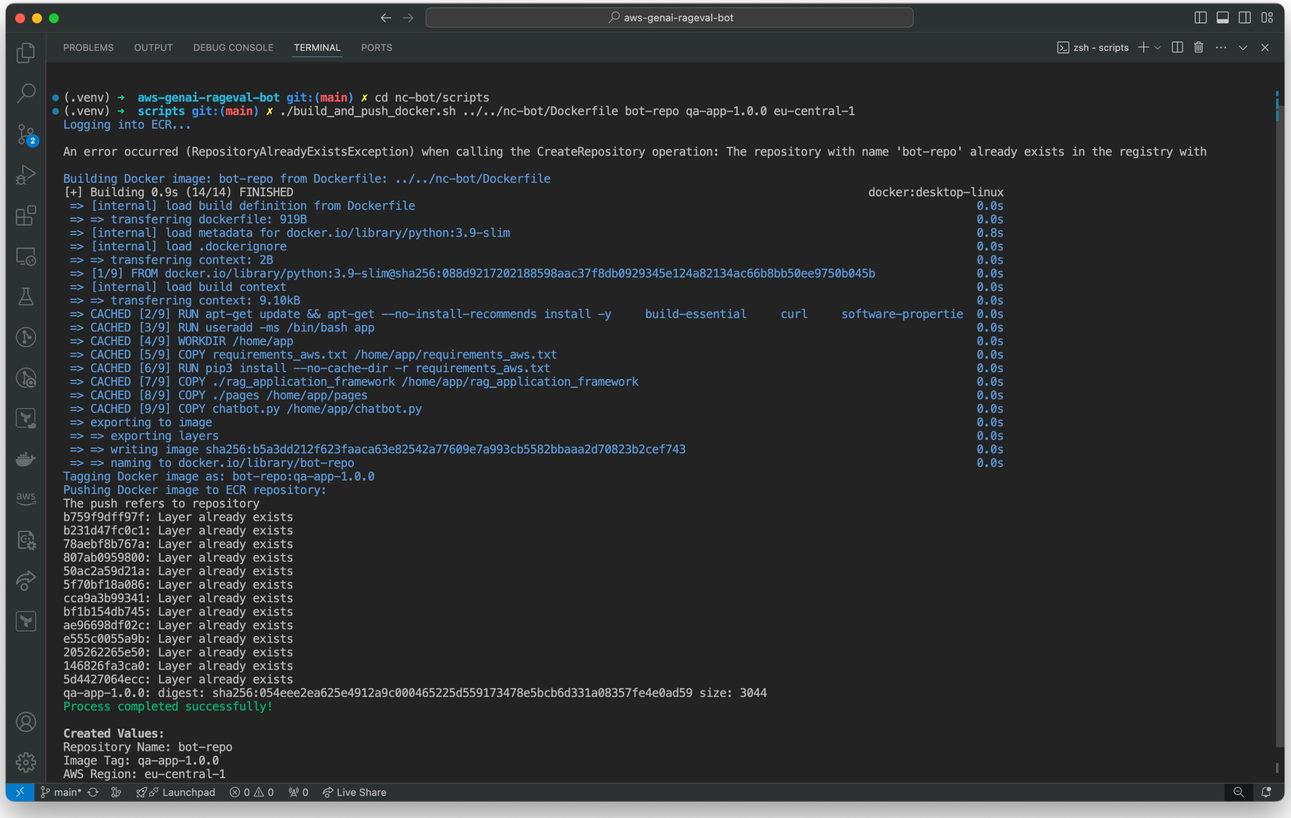
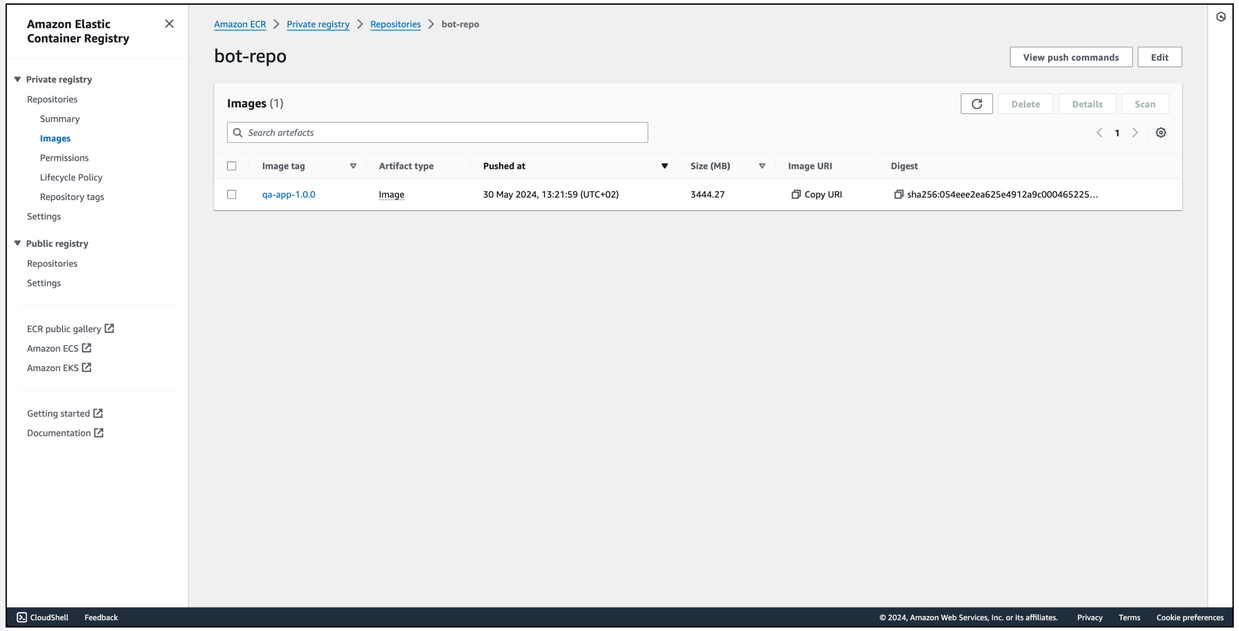
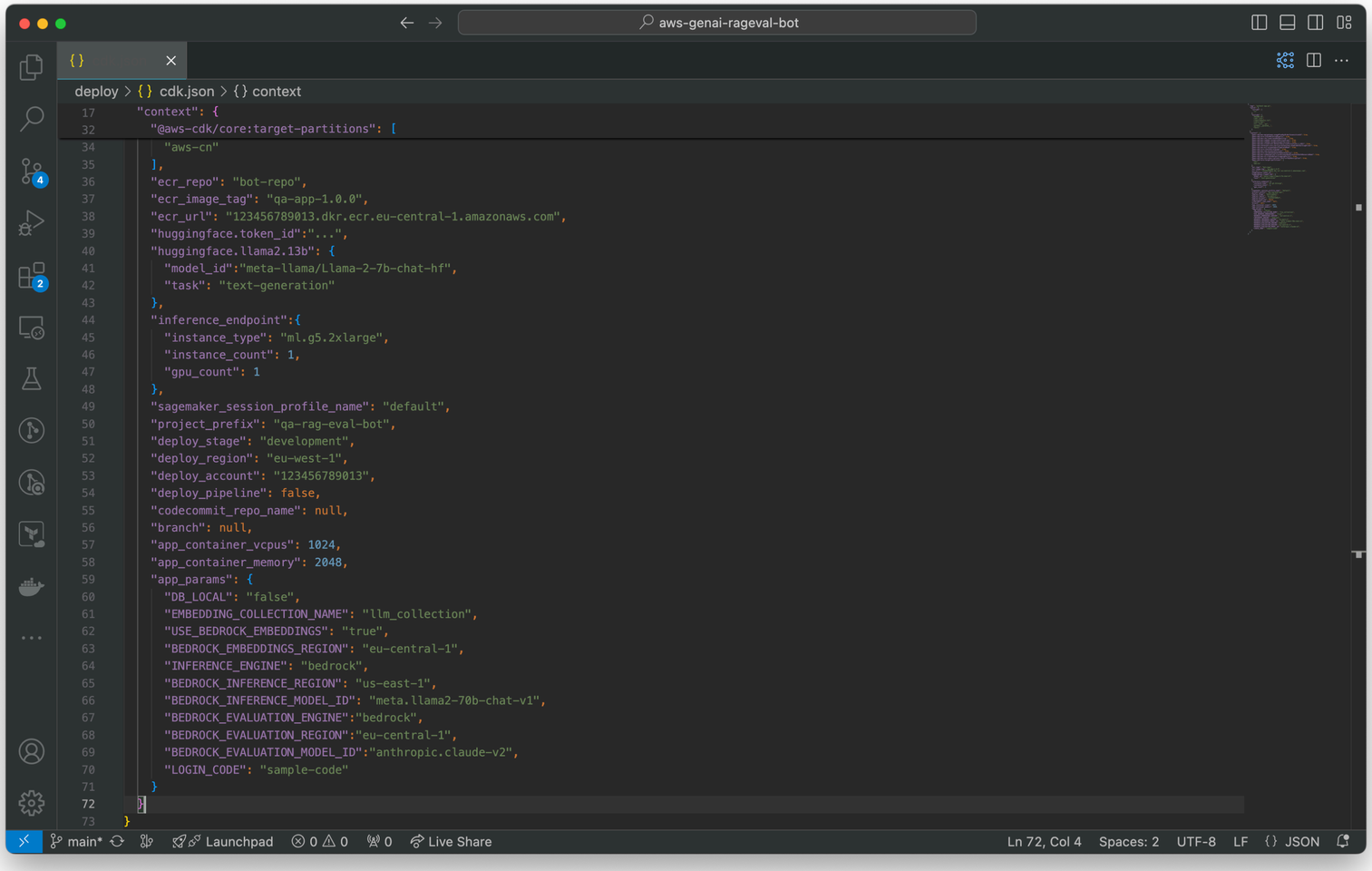
Docker Build and Push

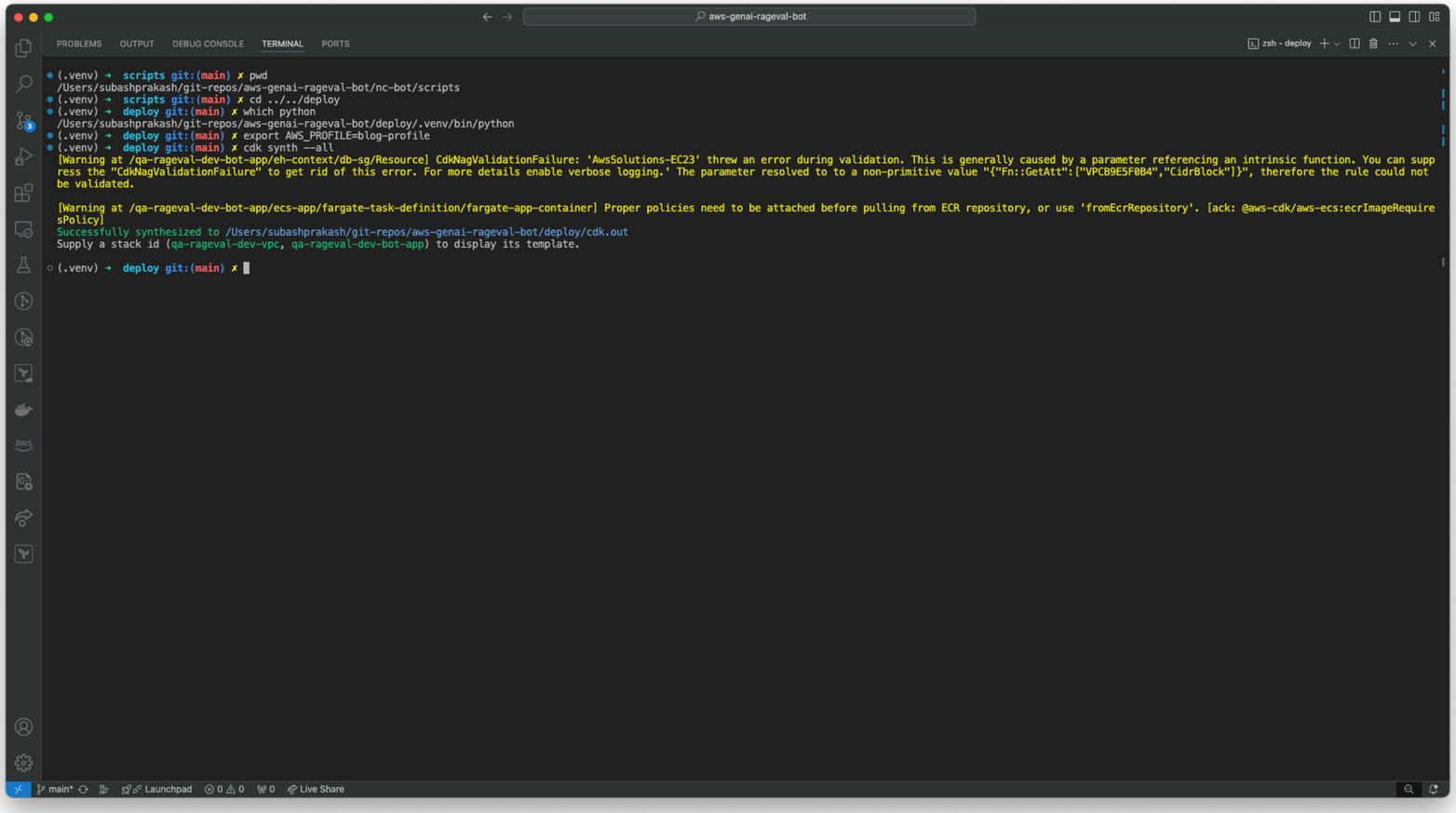




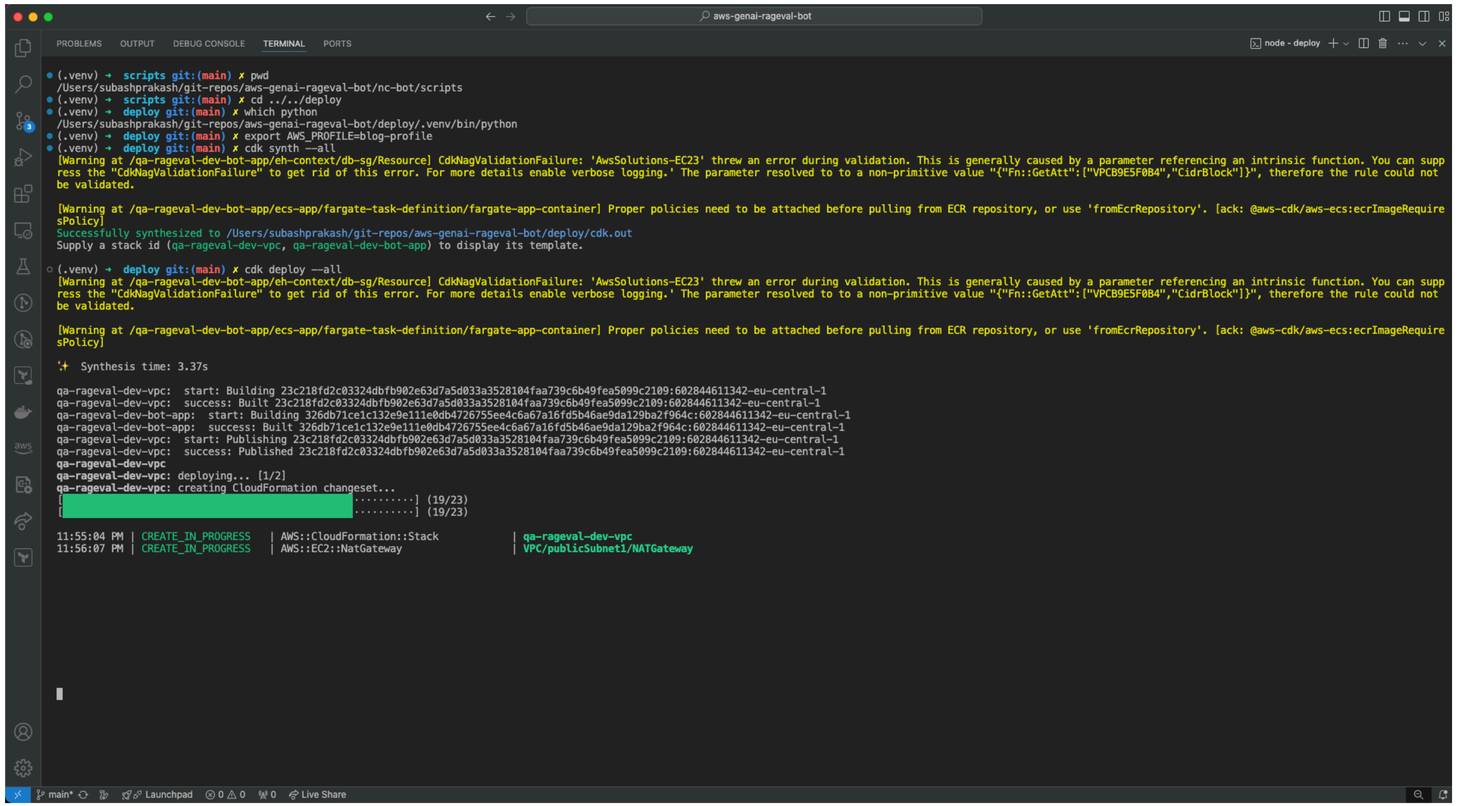
cdk.json example

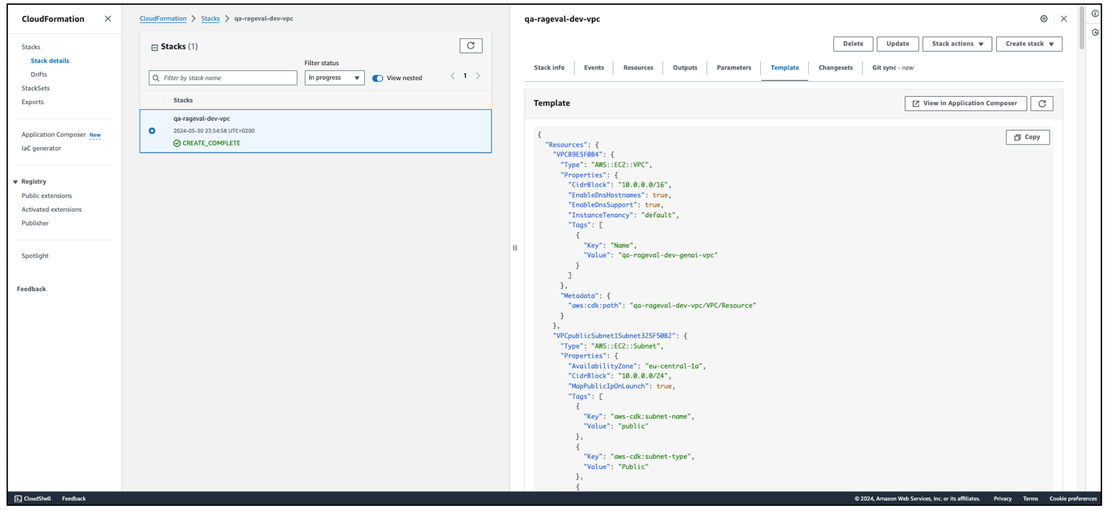


CDK synthesize



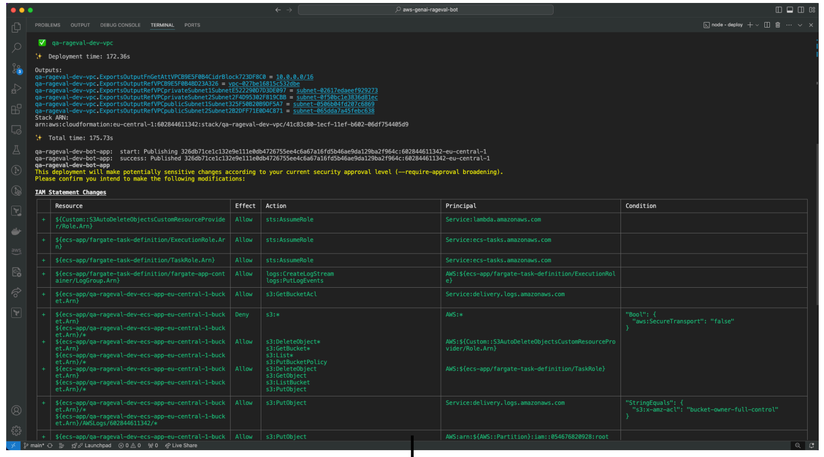
CDK deploy VPC Stack

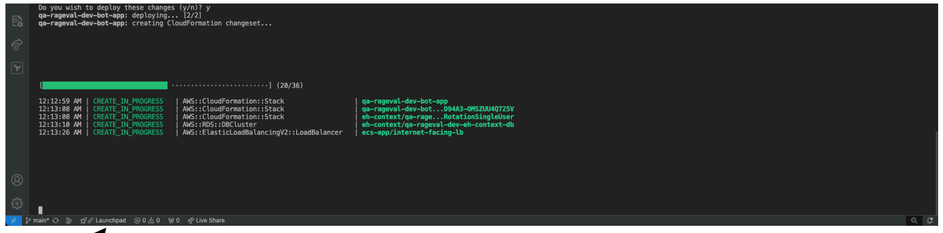




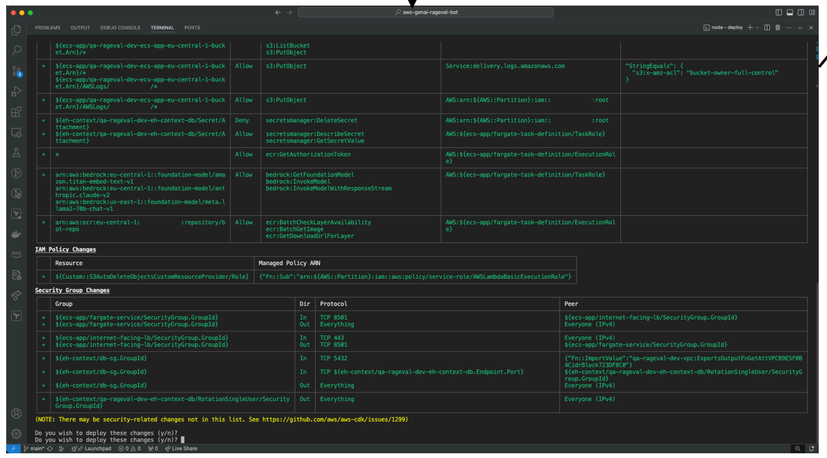
CDK deploy App Stack - 1

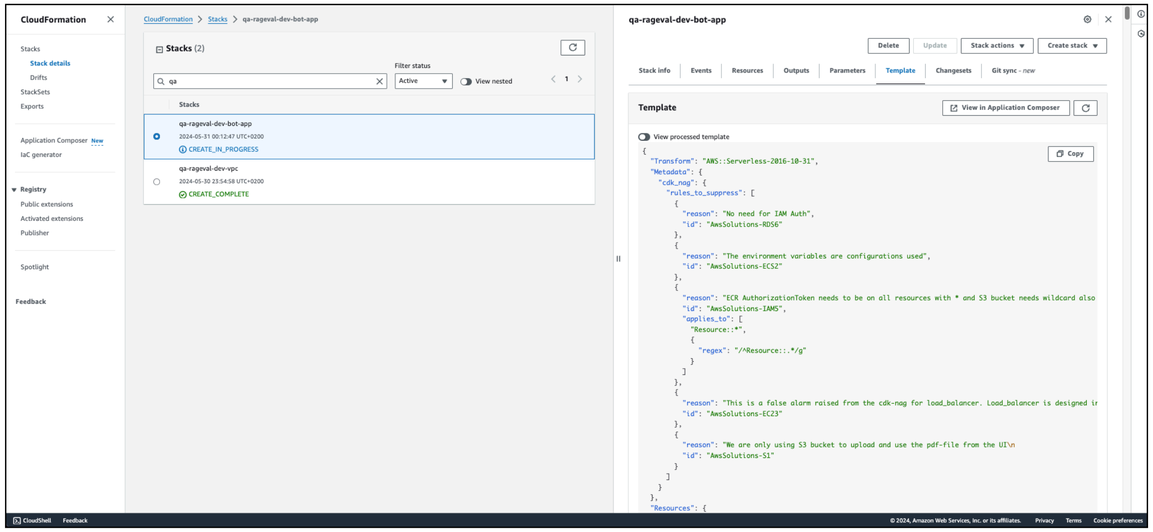
- The steps of the app stack deployment including the progress on CloudFormation Console.





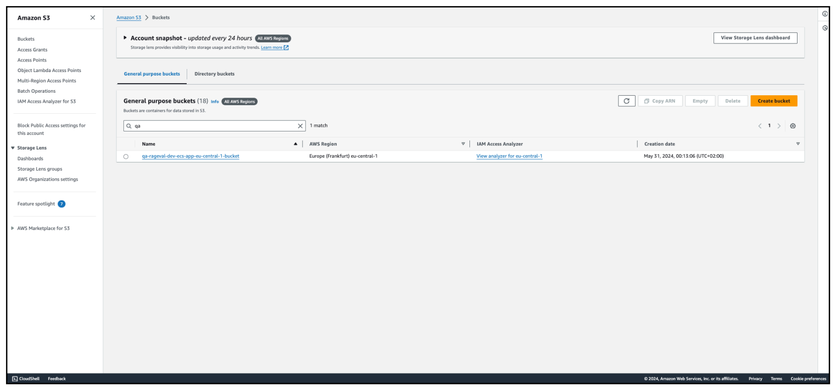
- Screenshot of the progress from the CloudFormation console.

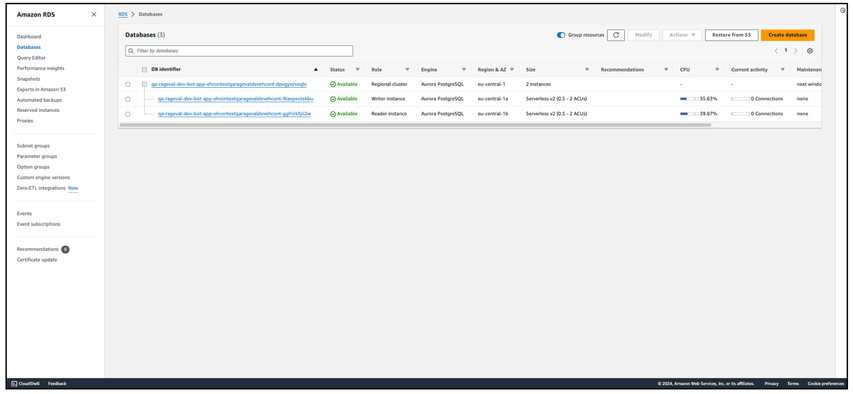


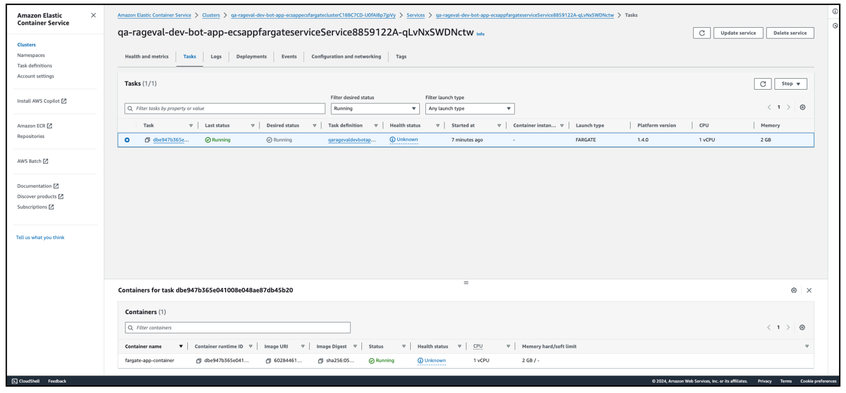


CDK deploy App Stack - 2

- The completion of the App Stack and verification of the resources including AWS S3, RDS, ECS

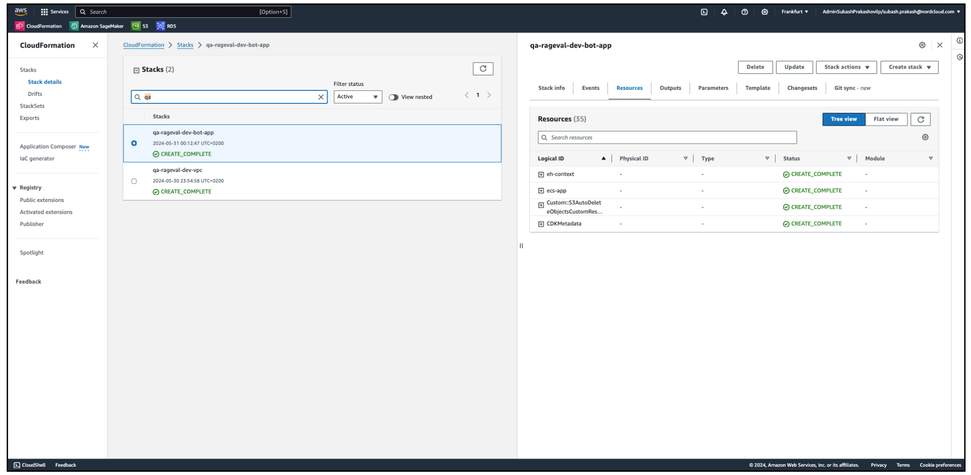




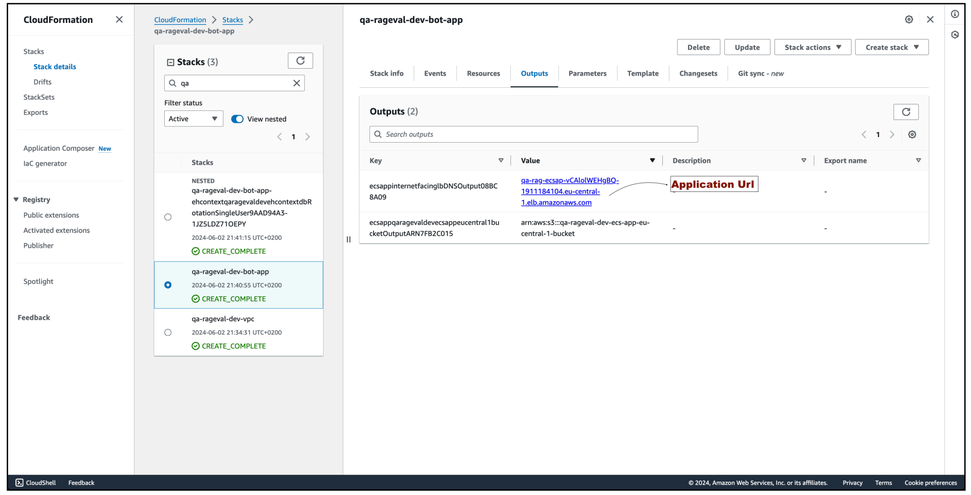


Accessing the Application

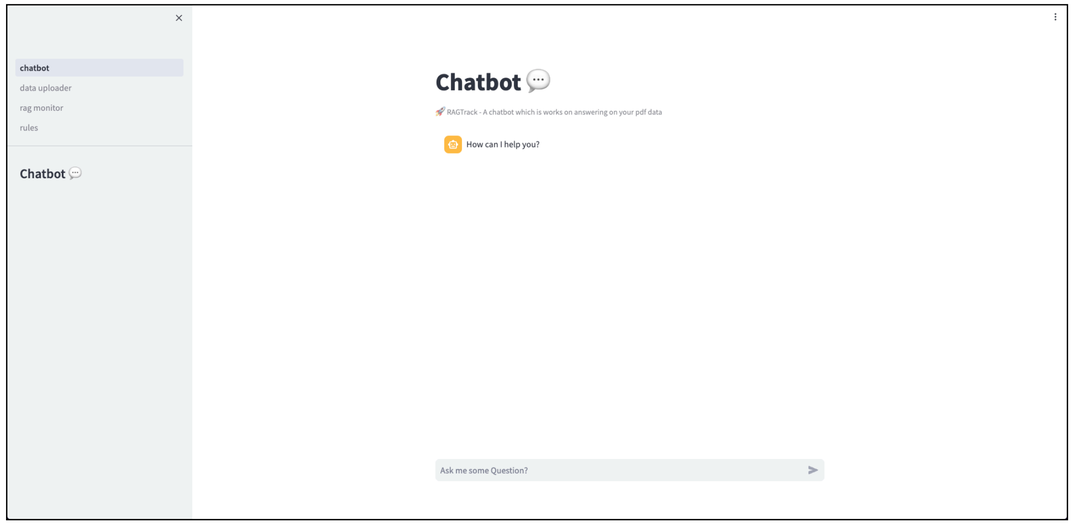
- Click on the app stack from CloudFormation Console, and select Outputs Tab from the right.



- Open the Application URL from the Outputs Tab to access the application

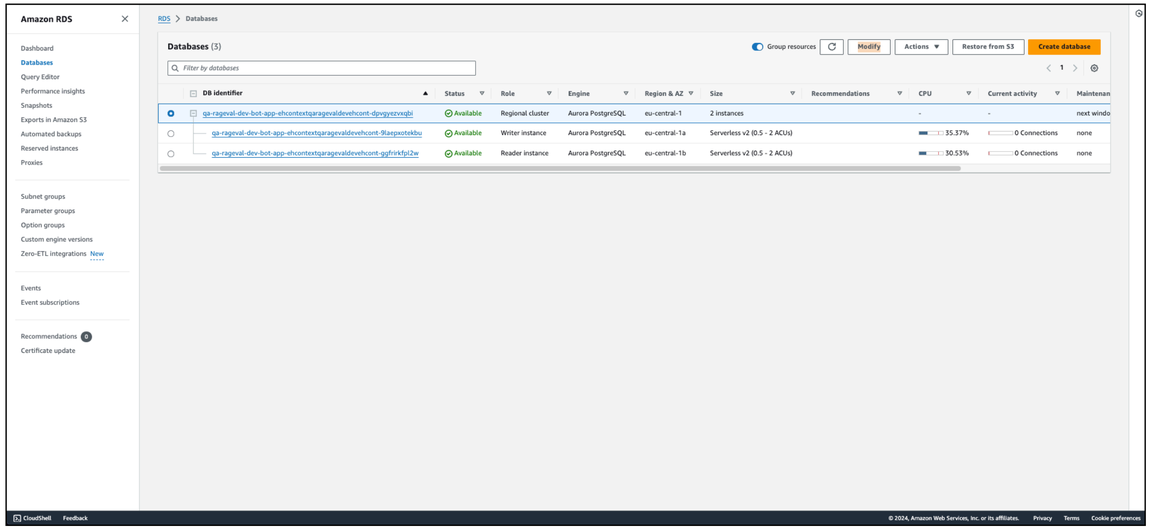


- Screenshot of the application

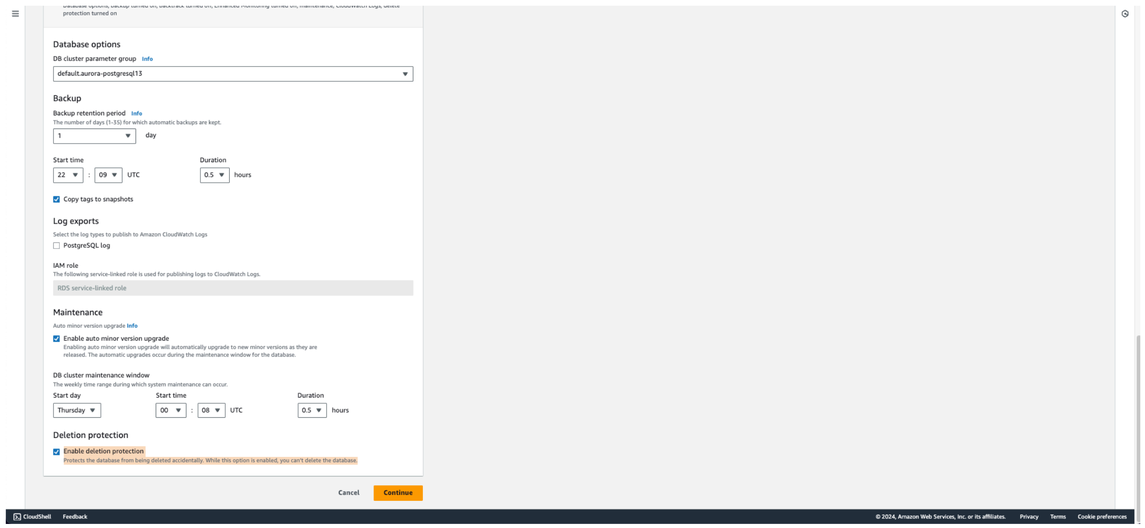


Cleaning up Prerequisites - RDS

- From AWS Console navigate to RDS —> Databases. Select the identified database and click on Modify as seen in the screenshot.

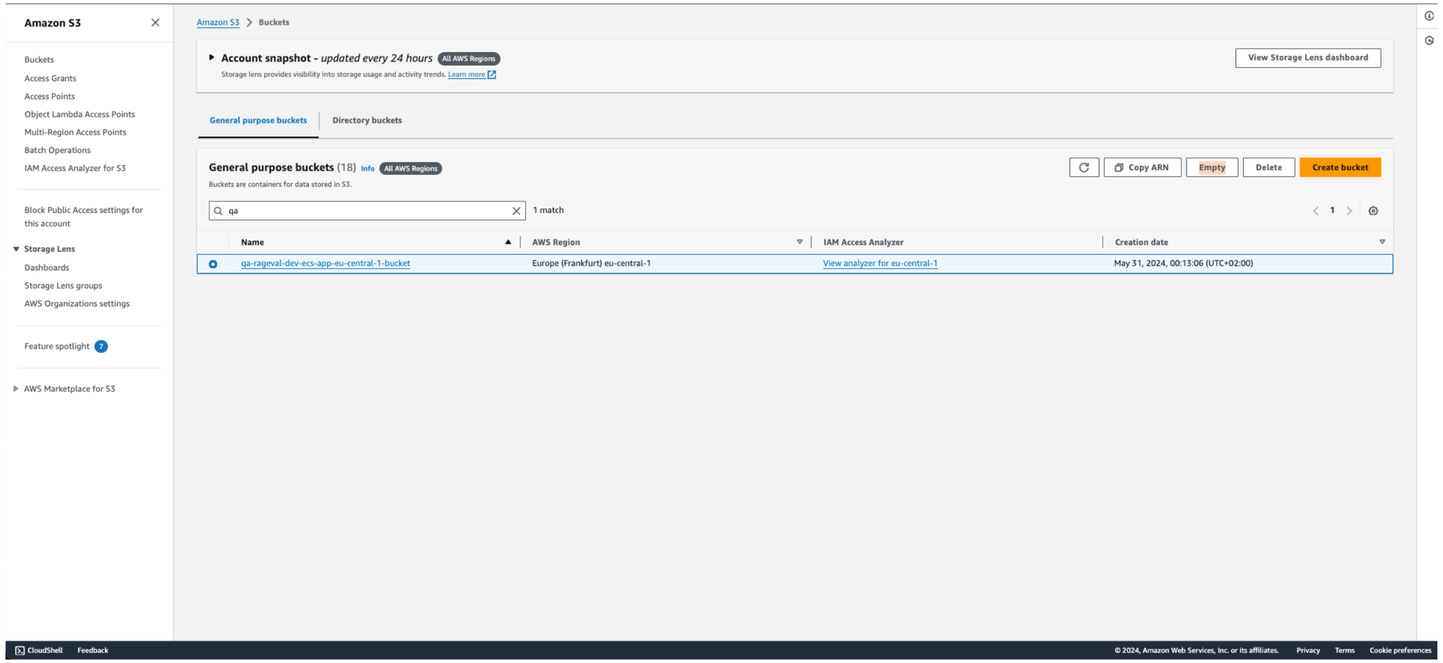


- Scroll to the end and untick the Enable Delete Protection. This will allow the cdk stack to be destroy since the database is delete protected according to the best practises.

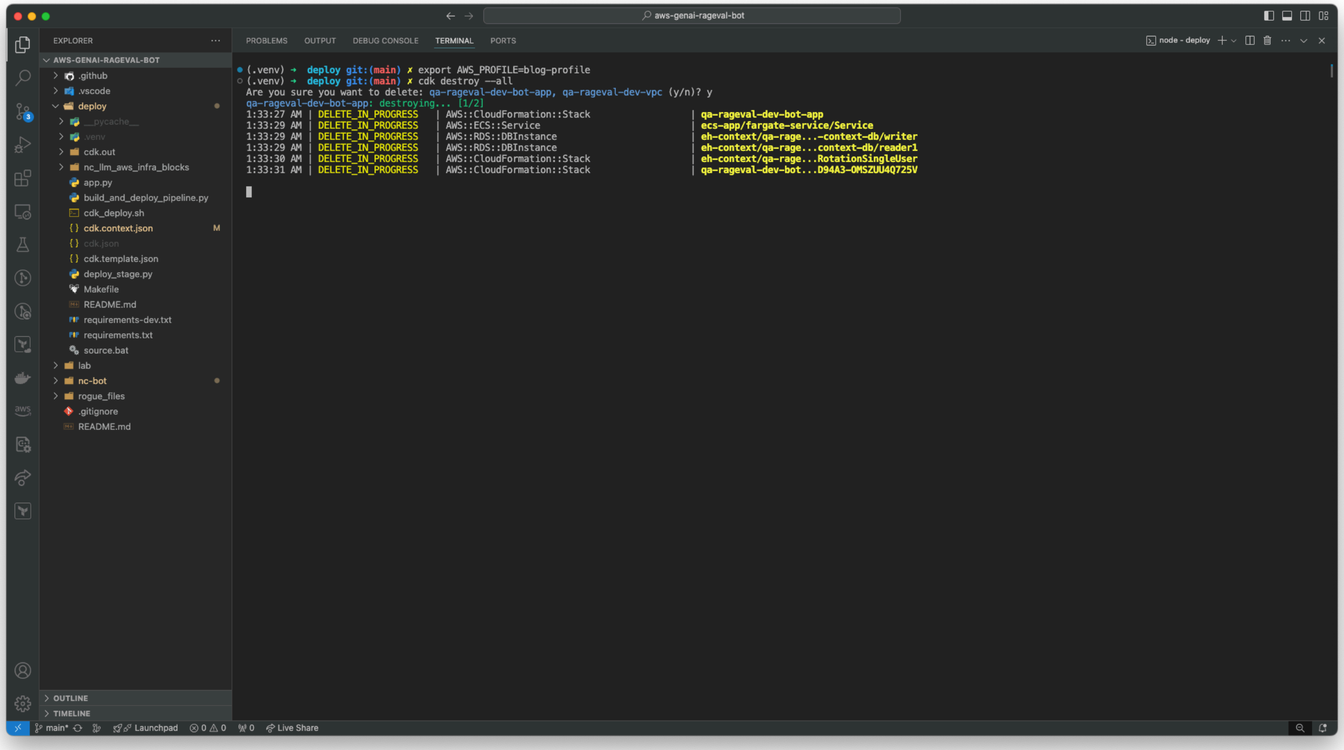


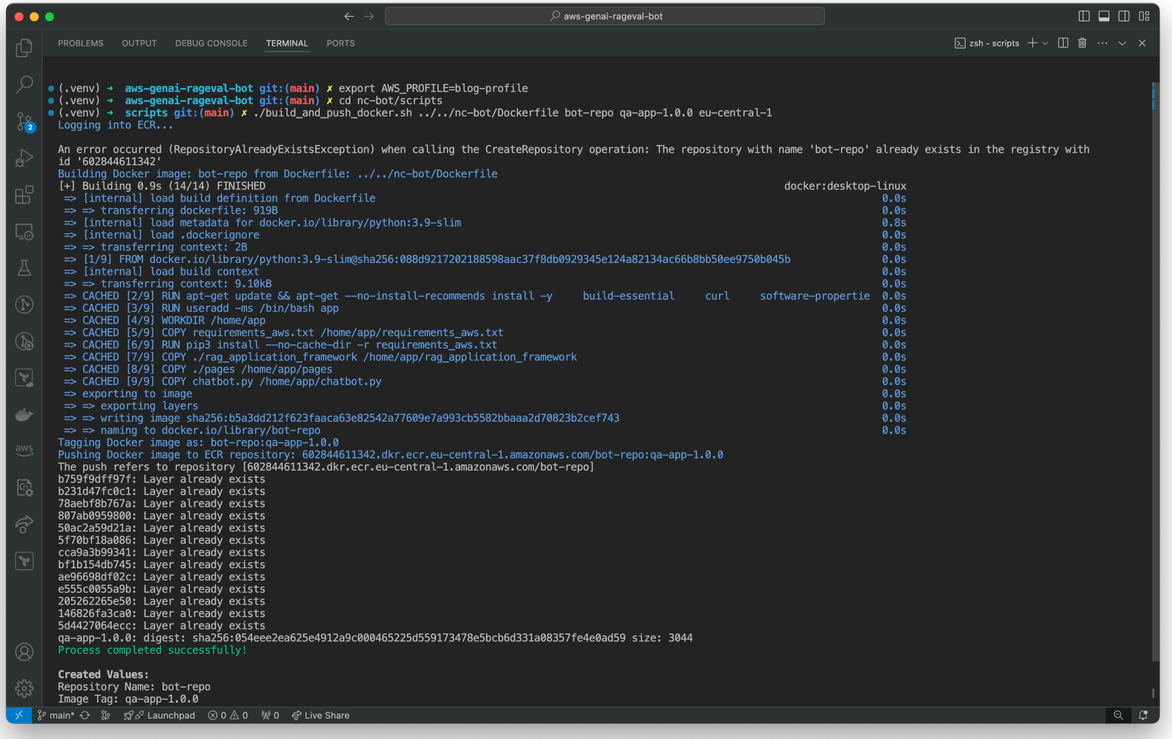
Cleaning up Prerequisites - S3

- From AWS Console navigate to S3. Select on the identified bucket, empty and delete it. An example is show below in the screenshot.



Cleaning up


aws-genai-rageval-bot
EXPLORER
AWS-GENAI-RAGEVAL-BOT
.github
.vscode
deploy
__pycache__
.venv
cdk.out
nc_llm_aws_infra_blocks
app.py
build_and_deploy_pipeline.py
cdk_deploy.sh
cdk.context.json
cdk.json
cdk.template.json
deploy_stage.py
Makefile
README.md
requirements-dev.txt
requirements.txt
source.bat
lab
nc-bot
rogue_files
.gitignore
README.md
OUTLINE
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node - deploy
(.venv) ➜ deploy git:(main) ✗ export AWS_PROFILE=blog-profile
(.venv) ➜ deploy git:(main) ✗ cdk destroy --all
Are you sure you want to delete: qa-rageval-dev-bot-app, qa-rageval-dev-vpc (y/n)? y
qa-rageval-dev-bot-app: destroying... [1/2]
1:33:27 AM | DELETE_IN_PROGRESS | AWS::CloudFormation::Stack | qa-rageval-dev-bot-app
1:33:29 AM | DELETE_IN_PROGRESS | AWS::ECS::Service | ecs-app/fargate-service/Service
1:33:29 AM | DELETE_IN_PROGRESS | AWS::RDS::DBInstance | eh-context/qa-rage...-context-db/writer
1:33:29 AM | DELETE_IN_PROGRESS | AWS::RDS::DBInstance | eh-context/qa-rage...context-db/reader1
1:33:30 AM | DELETE_IN_PROGRESS | AWS::CloudFormation::Stack | eh-context/qa-rage...RotationSingleUser
1:33:31 AM | DELETE_IN_PROGRESS | AWS::CloudFormation::Stack | qa-rageval-dev-bot...D94A3-OMSZUU4Q725V
main*
Launchpad
Live Share


aws-genai-rageval-bot
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
zsh - scripts
(.venv) ➜ aws-genai-rageval-bot git:(main) ✗ export AWS_PROFILE=blog-profile
(.venv) ➜ aws-genai-rageval-bot git:(main) ✗ cd nc-bot/scripts
(.venv) ➜ scripts git:(main) ✗ ./build_and_push_docker.sh ../../nc-bot/Dockerfile bot-repo qa-app-1.0.0 eu-central-1
Logging into ECR...
An error occurred (RepositoryAlreadyExistsException) when calling the CreateRepository operation: The repository with name 'bot-repo' already exists in the registry with id '602844611342'
Building Docker image: bot-repo from Dockerfile: ../../nc-bot/Dockerfile
[+] Building 0.9s (14/14) FINISHED docker:desktop-linux
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 919B 0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/9] FROM docker.io/library/python:3.9-slim@sha256:088d9217202188598aac37f8db0929345e124a82134ac66b8bb50ee9750b045b 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 9.10kB 0.0s
=> CACHED [2/9] RUN apt-get update && apt-get --no-install-recommends install -y build-essential curl software-propertie 0.0s
=> CACHED [3/9] RUN useradd -ms /bin/bash app 0.0s
=> CACHED [4/9] WORKDIR /home/app 0.0s
=> CACHED [5/9] COPY requirements_aws.txt /home/app/requirements_aws.txt 0.0s
=> CACHED [6/9] RUN pip3 install --no-cache-dir -r requirements_aws.txt 0.0s
=> CACHED [7/9] COPY ./rag_application_framework /home/app/rag_application_framework 0.0s
=> CACHED [8/9] COPY ./pages /home/app/pages 0.0s
=> CACHED [9/9] COPY chatbot.py /home/app/chatbot.py 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:b5a3dd212f623faaca63e82542a77609e7a993cb5582bbaaa2d70823b2cef743 0.0s
=> => naming to docker.io/library/bot-repo 0.0s
Tagging Docker image as: bot-repo:qa-app-1.0.0
Pushing Docker image to ECR repository: 602844611342.dkr.ecr.eu-central-1.amazonaws.com/bot-repo:qa-app-1.0.0
The push refers to repository [602844611342.dkr.ecr.eu-central-1.amazonaws.com/bot-repo]
b759f9dff97f: Layer already exists
b231d47fc0c1: Layer already exists
78aebf8b767a: Layer already exists
807ab0959800: Layer already exists
50ac2a59d21a: Layer already exists
5f70bf18a086: Layer already exists
cca9a3b99341: Layer already exists
bf1b154db745: Layer already exists
ae96698df02c: Layer already exists
e555c0055a9b: Layer already exists
205262265e50: Layer already exists
146826fa3ca0: Layer already exists
5d4427064ecc: Layer already exists
qa-app-1.0.0: digest: sha256:054eee2ea625e4912a9c000465225d559173478e5bcb6d331a08357fe4e0ad59 size: 3044
Process completed successfully!
Created Values:
Repository Name: bot-repo
Image Tag: qa-app-1.0.0
main*
Launchpad
Live Share