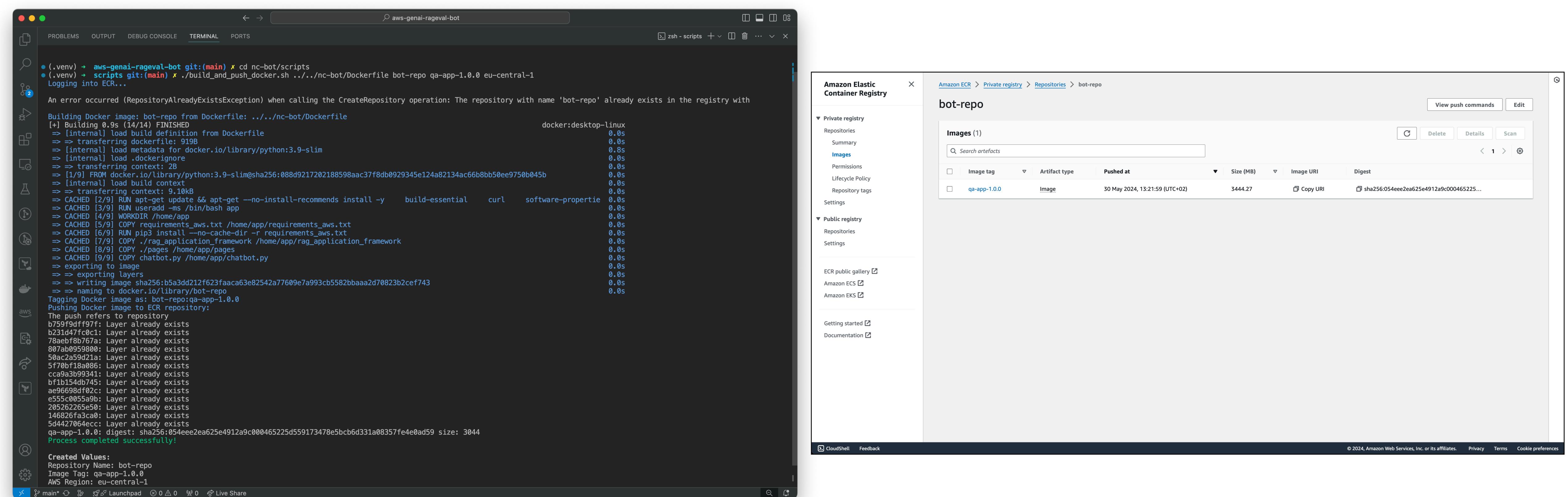


Docker Build and Push



The image shows two side-by-side screenshots illustrating the process of building and pushing a Docker image.

Terminal Screenshot:

```
(.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
Building Docker image: bot-repo from Dockerfile: ../../nc-bot/Dockerfile
[+] Building 0.9s (14/14) FINISHED
  => [internal] load build definition from Dockerfile
  => => transferring dockerfile: 919B
  => [internal] load metadata for docker.io/library/python:3.9-slim
  => [internal] load .dockerignore
  => => transferring context: 2B
  => [1/9] FROM docker.io/library/python:3.9-slim@sha256:088d9217202188598aac37f8db0929345e124a82134ac66b8bb50ee9750b045b
  => [internal] load build context
  => => transferring context: 9.10kB
  => CACHED [2/9] RUN apt-get update && apt-get --no-install-recommends install -y      build-essential      curl      software-properties
  => CACHED [3/9] RUN useradd -ms /bin/bash app
  => CACHED [4/9] WORKDIR /home/app
  => CACHED [5/9] COPY requirements_aws.txt /home/app/requirements_aws.txt
  => CACHED [6/9] RUN pip3 install --no-cache-dir -r requirements_aws.txt
  => CACHED [7/9] COPY ./rag_application_framework /home/app/rag_application_framework
  => CACHED [8/9] COPY ./pages /home/app/pages
  => CACHED [9/9] COPY chatbot.py /home/app/chatbot.py
  => exporting to image
  => => exporting layers
  => => writing image sha256:b5a3dd212f623faaca63e82542a77609e7a993cb5582bbaaa2d70823b2cef743
  => => naming to docker.io/library/bot-repo
Tagging Docker image as: bot-repo:qa-app-1.0.0
Pushing Docker image to ECR repository:
The push refers to repository
b759f9dff97f: Layer already exists
b231d47fc01: Layer already exists
78aebfb8b767a: 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:

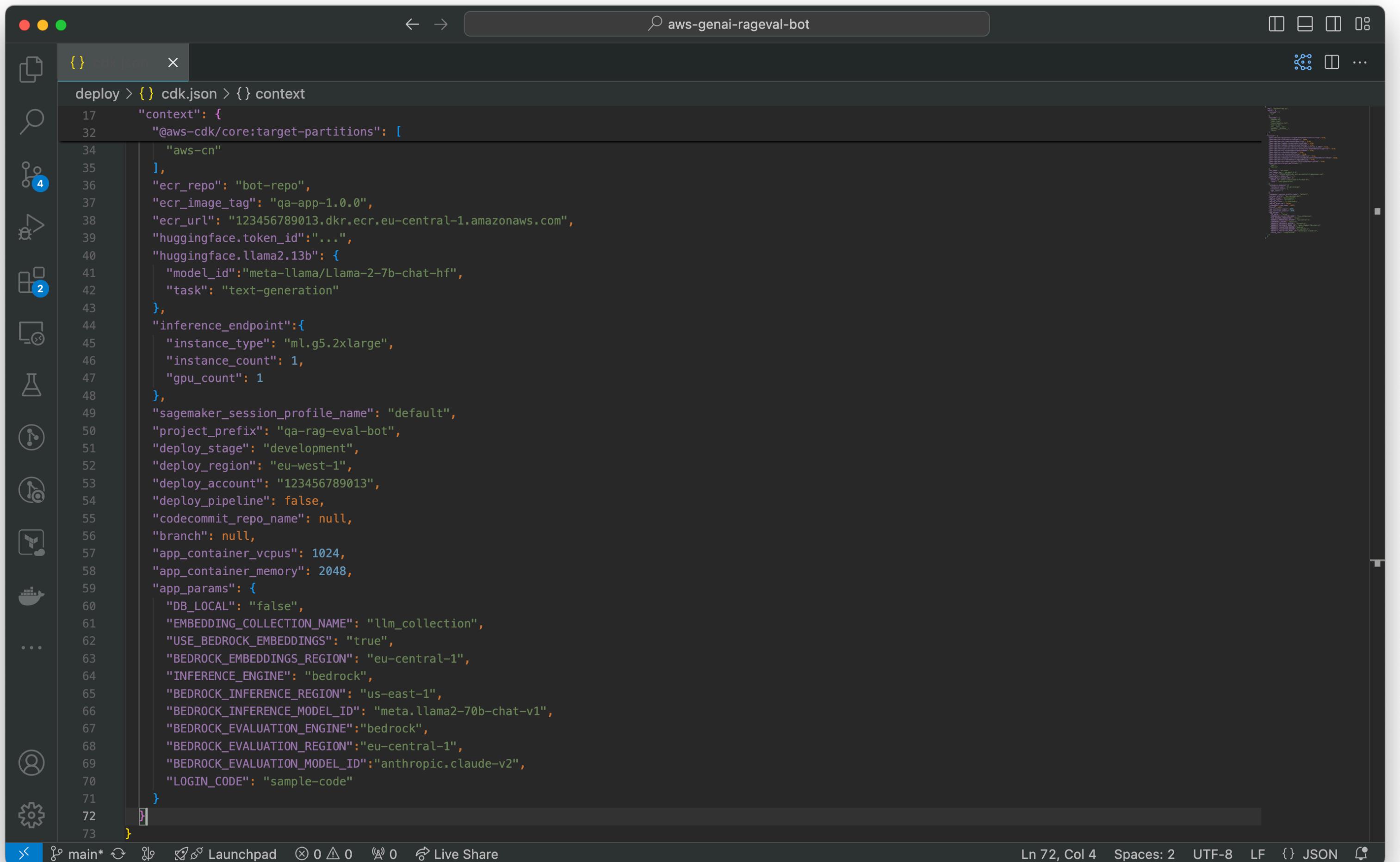
- Repository Name: bot-repo
- Image Tag: qa-app-1.0.0
- AWS Region: eu-central-1

Amazon Elastic Container Registry (ECR) Console Screenshot:

The screenshot shows the AWS ECR console with the 'bot-repo' repository selected. The 'Images' section displays one image entry:

Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest
qa-app-1.0.0	Image	30 May 2024, 13:21:59 (UTC+02)	3444.27	Copy URI	sha256:054eee2ea625e4912a9c000465225d559173478e5bcb6d331a08357fe4e0ad59...

cdk.json example



The screenshot shows a code editor window with a dark theme, displaying a `cdk.json` configuration file. The file is titled `aws-genai-rageval-bot` and contains JSON code for deploying a machine learning application. The code defines various parameters such as target partitions, ECR repository details, inference endpoint specifications, and deployment configurations for Sagemaker and Bedrock.

```
deploy > {} cdk.json > {} context
  "context": {
    "@aws-cdk/core:target-partitions": [
      "aws-cn"
    ],
    "ecr_repo": "bot-repo",
    "ecr_image_tag": "qa-app-1.0.0",
    "ecr_url": "123456789013.dkr.ecr.eu-central-1.amazonaws.com",
    "huggingface.token_id": "...",
    "huggingface.llama2.13b": {
      "model_id": "meta-llama/Llama-2-7b-chat-hf",
      "task": "text-generation"
    },
    "inference_endpoint": {
      "instance_type": "ml.g5.2xlarge",
      "instance_count": 1,
      "gpu_count": 1
    },
    "sagemaker_session_profile_name": "default",
    "project_prefix": "qa-rag-eval-bot",
    "deploy_stage": "development",
    "deploy_region": "eu-west-1",
    "deploy_account": "123456789013",
    "deploy_pipeline": false,
    "codecommit_repo_name": null,
    "branch": null,
    "app_container_vcpus": 1024,
    "app_container_memory": 2048,
    "app_params": {
      "DB_LOCAL": "false",
      "EMBEDDING_COLLECTION_NAME": "llm_collection",
      "USE_BEDROCK_EMBEDDINGS": "true",
      "BEDROCK_EMBEDDINGS_REGION": "eu-central-1",
      "INFERENCE_ENGINE": "bedrock",
      "BEDROCK_INFERENCE_REGION": "us-east-1",
      "BEDROCK_INFERENCE_MODEL_ID": "meta.llama2-70b-chat-v1",
      "BEDROCK_EVALUATION_ENGINE": "bedrock",
      "BEDROCK_EVALUATION_REGION": "eu-central-1",
      "BEDROCK_EVALUATION_MODEL_ID": "anthropic.claude-v2",
      "LOGIN_CODE": "sample-code"
    }
  }
}
```

The editor interface includes a sidebar with various icons for file operations, a status bar at the bottom showing file path, line number, and encoding, and a floating preview window in the background showing the visual representation of the deployed resources.

CDK synthesize

The screenshot shows a terminal window in the VS Code interface. The title bar of the terminal says "aws-genai-rageval-bot". The terminal content displays a series of commands run in a virtual environment (.venv) and their corresponding outputs:

- (.venv) + scripts git:(main) x pwd
/Users/subashprakash/git-repos/aws-genai-rageval-bot/nc-bot/scripts
- (.venv) + scripts git:(main) x cd ../../deploy
- (.venv) + deploy git:(main) x which python
/Users/subashprakash/git-repos/aws-genai-rageval-bot/deploy/.venv/bin/python
- (.venv) + deploy git:(main) x export AWS_PROFILE=blog-profile
- (.venv) + deploy git:(main) x cdk synth --all
[Warning at /qa-rageval-dev-bot-app/eh-context/db-sg/Resource] CdkNagValidationFailure: 'AwsSolutions-EC23' threw an error during validation. This is generally caused by a parameter referencing an intrinsic function. You can suppress the "CdkNagValidationFailure" to get rid of this error. For more details enable verbose logging.' The parameter resolved to a non-primitive value "{\"Fn::GetAtt\": [\"VPCB9E5F0B4\", \"CidrBlock\"]}", therefore the rule could not be validated.
- [Warning at /qa-rageval-dev-bot-app/ecs-app/fargate-task-definition/fargate-app-container] Proper policies need to be attached before pulling from ECR repository, or use 'fromEcrRepository'. [ack: @aws-cdk/aws-ecs:ecrImageRequiresPolicy]
- Successfully synthesized to /Users/subashprakash/git-repos/aws-genai-rageval-bot/deploy/cdk.out
Supply a stack id (qa-rageval-dev-vpc, qa-rageval-dev-bot-app) to display its template.
- (.venv) + deploy git:(main) x

CDK deploy VPC Stack

The screenshot shows a terminal window in VS Code with the title bar "aws-genai-rageval-bot". The terminal tab is selected, and the content displays the output of an AWS CDK deployment command. The logs include several warning messages about CdkNagValidationFailure due to intrinsic function references in CloudFormation templates. It also shows the synthesis time as 3.37s and the deployment of multiple stacks (qa-rageval-dev-vpc, qa-rageval-dev-bot-app) with their respective stack IDs and regions. A progress bar at the bottom indicates the creation of a CloudFormation changeset. The bottom status bar shows AWS CloudWatch log entries for the deployment process.

```
● (.venv) ➜ scripts git:(main) ✘ pwd
/Users/subashprakash/git-repos/aws-genai-rageval-bot/nc-bot/scripts
● (.venv) ➜ scripts git:(main) ✘ cd ../../deploy
● (.venv) ➜ deploy git:(main) ✘ which python
/Users/subashprakash/git-repos/aws-genai-rageval-bot/deploy/.venv/bin/python
● (.venv) ➜ deploy git:(main) ✘ export AWS_PROFILE=blog-profile
● (.venv) ➜ deploy git:(main) ✘ cdk synth --all
[Warning at /qa-rageval-dev-bot-app/reh-context/db-sg/Resource] CdkNagValidationFailure: 'AwsSolutions-EC23' threw an error during validation. This is generally caused by a parameter referencing an intrinsic function. You can suppress the "CdkNagValidationFailure" to get rid of this error. For more details enable verbose logging.' The parameter resolved to to a non-primitive value "{\"Fn::GetAtt\": [\"VPCB9E5F0B4\", \"CidrBlock\"]}", therefore the rule could not be validated.

[Warning at /qa-rageval-dev-bot-app/ecs-app/fargate-task-definition/fargate-app-container] Proper policies need to be attached before pulling from ECR repository, or use 'fromEcrRepository'. [ack: @aws-cdk/aws-ecs:ecrImageRequiresPolicy]
Successfully synthesized to /Users/subashprakash/git-repos/aws-genai-rageval-bot/deploy/cdk.out
Supply a stack id (qa-rageval-dev-vpc, qa-rageval-dev-bot-app) to display its template.

○ (.venv) ➜ deploy git:(main) ✘ cdk deploy --all
[Warning at /qa-rageval-dev-bot-app/reh-context/db-sg/Resource] CdkNagValidationFailure: 'AwsSolutions-EC23' threw an error during validation. This is generally caused by a parameter referencing an intrinsic function. You can suppress the "CdkNagValidationFailure" to get rid of this error. For more details enable verbose logging.' The parameter resolved to to a non-primitive value "{\"Fn::GetAtt\": [\"VPCB9E5F0B4\", \"CidrBlock\"]}", therefore the rule could not be validated.

[Warning at /qa-rageval-dev-bot-app/ecs-app/fargate-task-definition/fargate-app-container] Proper policies need to be attached before pulling from ECR repository, or use 'fromEcrRepository'. [ack: @aws-cdk/aws-ecs:ecrImageRequiresPolicy]

⚡ Synthesis time: 3.37s

qa-rageval-dev-vpc: start: Building 23c218fd2c03324dbfb902e63d7a5d033a3528104faa739c6b49fea5099c2109:602844611342-eu-central-1
qa-rageval-dev-vpc: success: Built 23c218fd2c03324dbfb902e63d7a5d033a3528104faa739c6b49fea5099c2109:602844611342-eu-central-1
qa-rageval-dev-bot-app: start: Building 326db71ce1c132e9e11le0db4726755ee4c6a67a16fd5b46ae9da129ba2f964c:602844611342-eu-central-1
qa-rageval-dev-bot-app: success: Built 326db71ce1c132e9e11le0db4726755ee4c6a67a16fd5b46ae9da129ba2f964c:602844611342-eu-central-1
qa-rageval-dev-vpc: start: Publishing 23c218fd2c03324dbfb902e63d7a5d033a3528104faa739c6b49fea5099c2109:602844611342-eu-central-1
qa-rageval-dev-vpc: success: Published 23c218fd2c03324dbfb902e63d7a5d033a3528104faa739c6b49fea5099c2109:602844611342-eu-central-1
qa-rageval-dev-vpc
qa-rageval-dev-vpc: deploying... [1/2]
qa-rageval-dev-vpc: creating CloudFormation changeset...
[.....] (19/23)
[.....] (19/23)

11:55:04 PM | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | qa-rageval-dev-vpc
11:56:07 PM | CREATE_IN_PROGRESS | AWS::EC2::NatGateway | VPC/publicSubnet1/NATGateway
```

The screenshot shows the AWS CloudFormation console interface. On the left, the navigation pane includes sections for CloudFormation, Stacks, Stack details, Drifts, StackSets, Exports, Application Composer (with a 'New' button), Registry (Public extensions, Activated extensions, Publisher), Spotlight, and Feedback.

The main area displays the 'qa-rageval-dev-vpc' stack details. The top navigation bar shows the path: CloudFormation > Stacks > qa-rageval-dev-vpc. The top right features standard actions: Delete, Update, Stack actions, and Create stack.

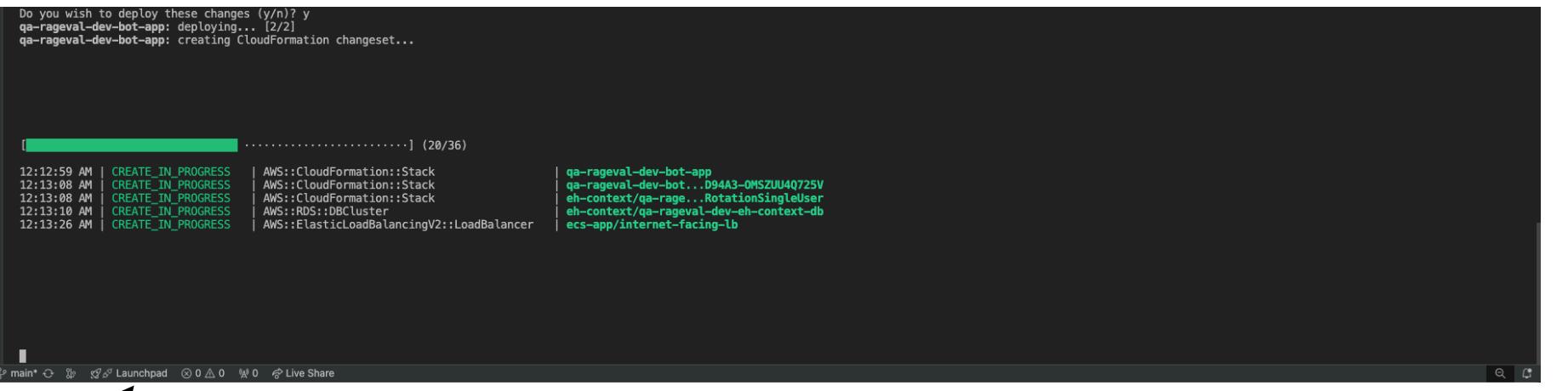
The 'Stacks (1)' section lists the single stack 'qa-rageval-dev-vpc' with the status 'CREATE_COMPLETE'. It was created on '2024-05-30 23:54:58 UTC+0200'. A 'View in Application Composer' button is available for this stack.

The 'Template' tab is selected, displaying the CloudFormation JSON template:

```
{ "Resources": { "VPCB9E5F0B4": { "Type": "AWS::EC2::VPC", "Properties": { "CidrBlock": "10.0.0.0/16", "EnableDnsHostnames": true, "EnableDnsSupport": true, "InstanceTenancy": "default", "Tags": [ { "Key": "Name", "Value": "qa-rageval-dev-genai-vpc" } ] }, "Metadata": { "aws:cdk:path": "qa-rageval-dev-vpc/VPC/Resource" } }, "VPCpublicSubnet1Subnet325F50B2": { "Type": "AWS::EC2::Subnet", "Properties": { "AvailabilityZone": "eu-central-1a", "CidrBlock": "10.0.0.0/24", "MapPublicIpOnLaunch": true, "Tags": [ { "Key": "aws-cdk:subnet-name", "Value": "public" }, { "Key": "aws-cdk:subnet-type", "Value": "Public" } ] } } }
```

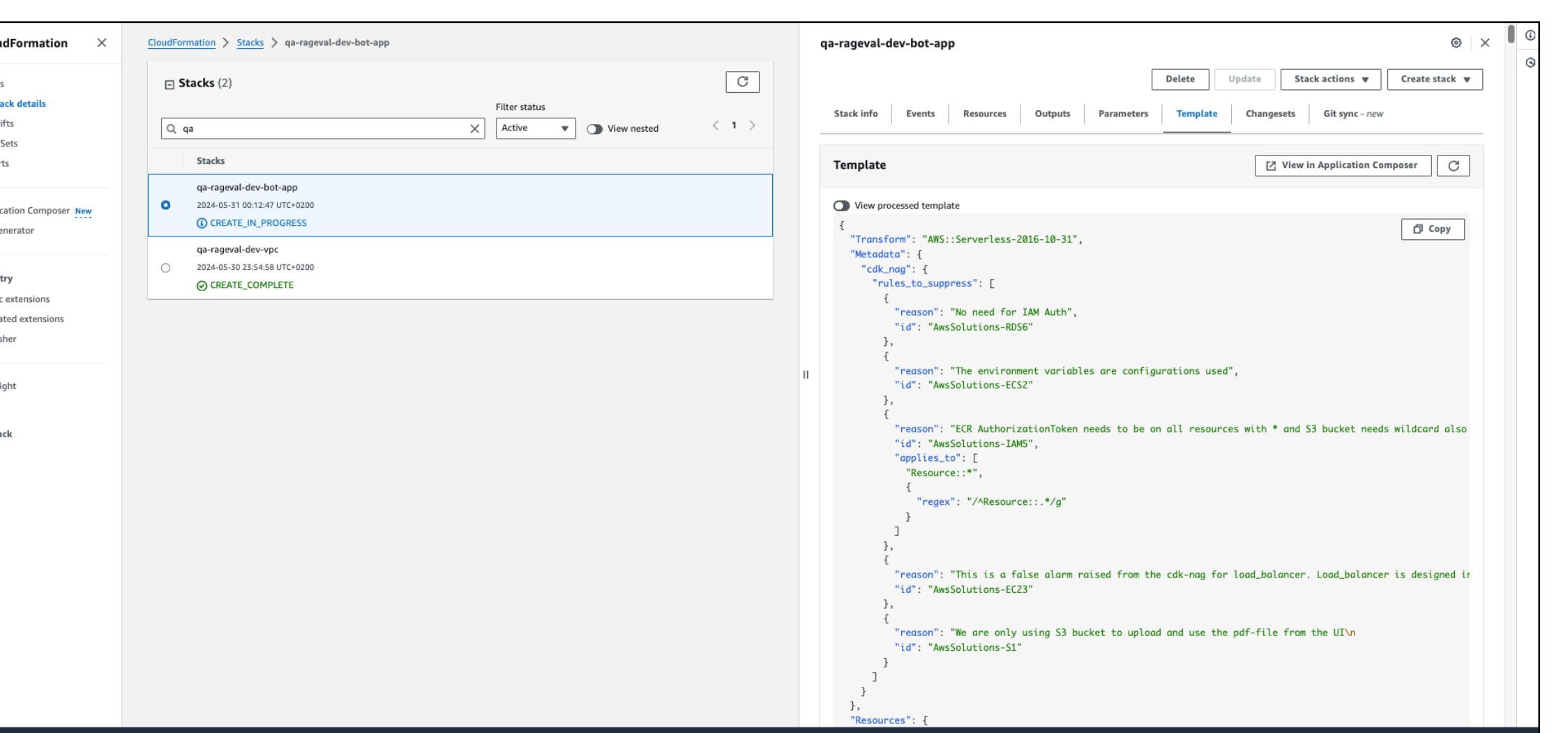
CDK deploy App Stack - 1

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



A screenshot of the AWS CloudFormation console showing the progress of a stack named "qa-rageval-dev-bot-app". The status bar at the bottom indicates "CREATE_IN_PROGRESS" and "2/2 resources". The main area shows two resources: "qa-rageval-dev-bot-app" and "qa-rageval-dev-vpc". The "qa-rageval-dev-vpc" resource has a status of "CREATE_IN_PROGRESS". The "qa-rageval-dev-bot-app" resource has a status of "CREATE_COMPLETE". A large arrow points from the top-left towards this screenshot.

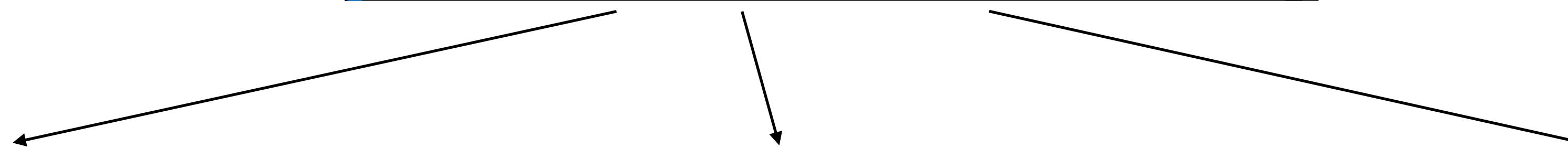
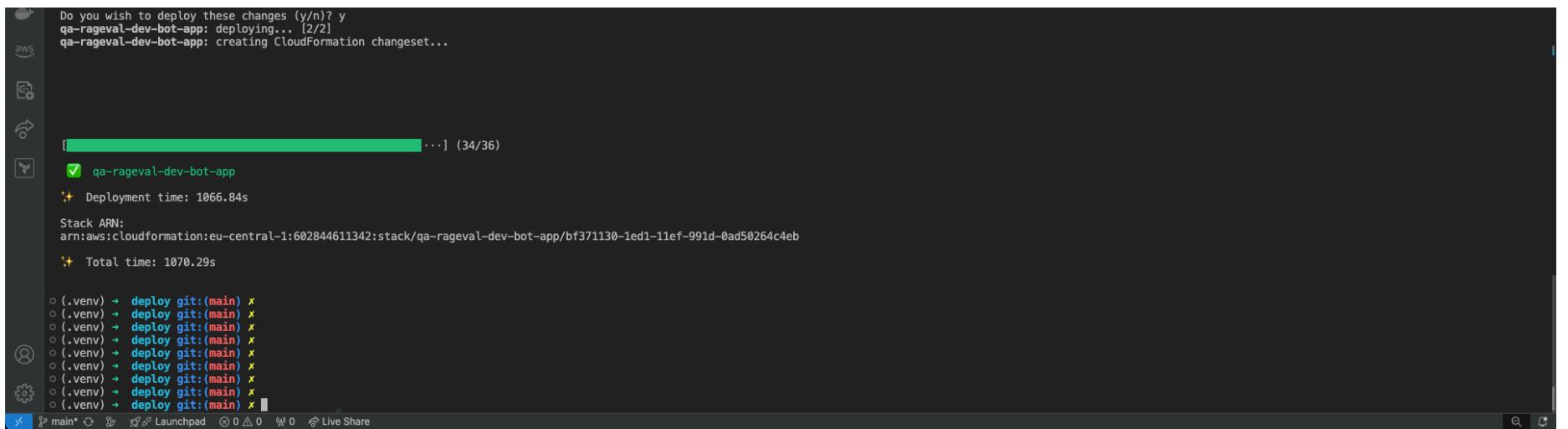
- Screenshot of the progress from the CloudFormation console.



A screenshot of the AWS CloudFormation console showing the progress of a stack named "qa-rageval-dev-bot-app". The status bar at the bottom indicates "CREATE_IN_PROGRESS" and "2/2 resources". The main area shows two resources: "qa-rageval-dev-bot-app" and "qa-rageval-dev-vpc". The "qa-rageval-dev-vpc" resource has a status of "CREATE_IN_PROGRESS". The "qa-rageval-dev-bot-app" resource has a status of "CREATE_COMPLETE". A large arrow points from the bottom-left towards this screenshot.

CDK deploy App Stack - 2

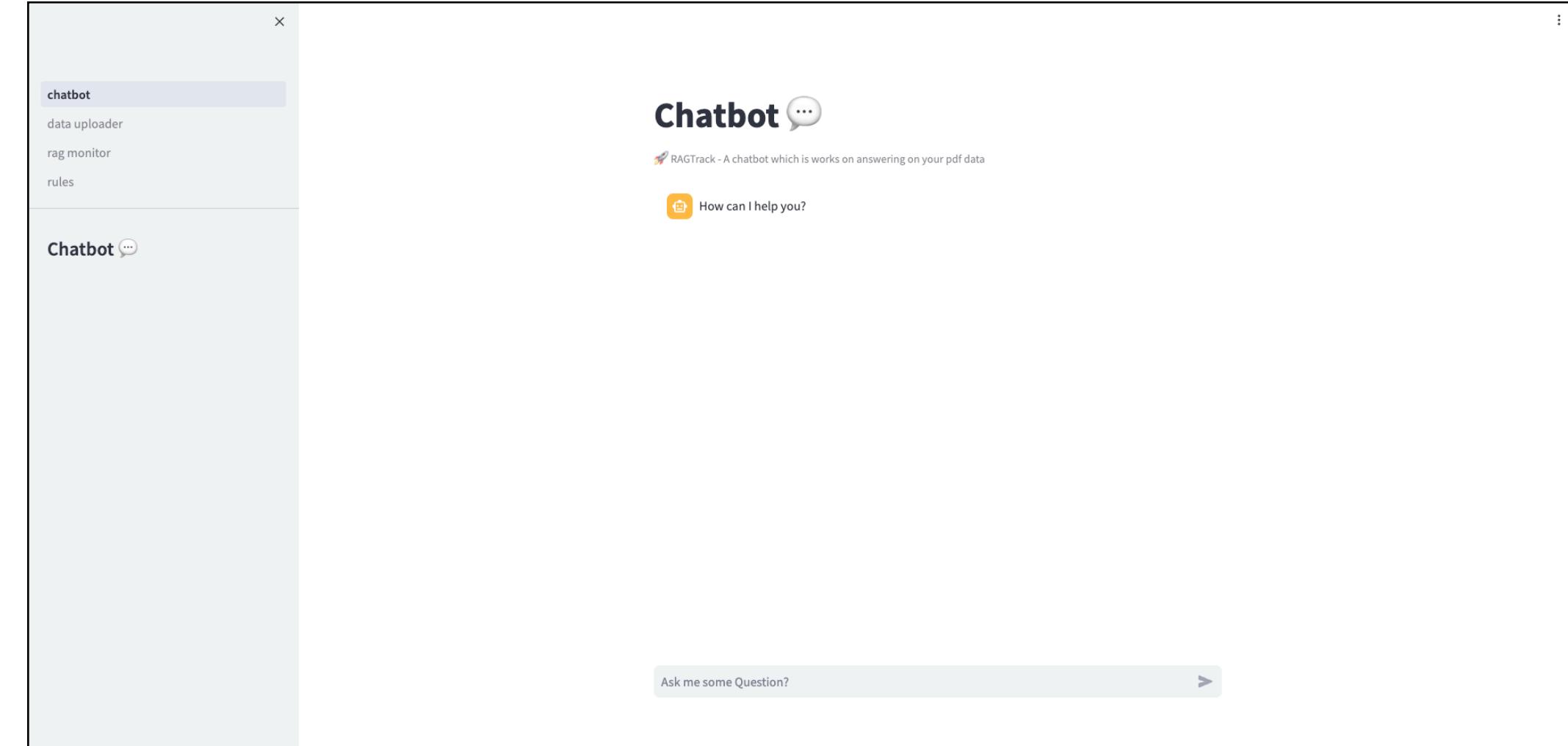
- The completion of the App Stack and verification of the resources including AWS S3, RDS, ECS



Accessing the Application

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

The screenshot shows the AWS CloudFormation console. On the left, there's a sidebar with various navigation options like Drifts, StackSets, Application Composer, Registry, and Feedback. The main area shows a list of stacks under 'Stacks (2)'. One stack is selected: 'qa-rageval-dev-bot-app' (2024-05-31 00:12:47 UTC+0200). Below it is another stack: 'qa-rageval-dev-vpc' (2024-05-30 23:54:58 UTC+0200). The 'Outputs' tab is currently selected, showing a table with one row: 'ecsappinternetfacinglbDNSOutput08C8A09' with a value of 'qa-rag-ecsap-vCAoIWEHgBQ-1911184104.eu-central-1.elb.amazonaws.com'.



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

This screenshot is similar to the previous one but focuses on the 'Outputs' tab for the 'qa-rageval-dev-bot-app' stack. The table shows two outputs: 'ecsappinternetfacinglbDNSOutput08C8A09' with the value 'qa-rag-ecsap-vCAoIWEHgBQ-1911184104.eu-central-1.elb.amazonaws.com' and 'ecsapppargevaldevcecsappecentral1bucketecketOutputARN7FB2C015' with the value 'arn:aws:s3:::qa-rageval-dev-ecs-app-eu-central-1-bucket'. The 'Value' column for the first output is highlighted in red with the label 'Application Url'.

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 destroyed since the database is delete protected according to the best practises.

The screenshot shows the AWS RDS Databases page. On the left is a sidebar with links like Dashboard, Databases, Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL Integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main area is titled 'Databases (3)' and lists three entries:

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU	Current activity	Maintenance
ga-rageval-dev-bot-app-ehcontextqaragevaldevehcont-dpvgvezxqb1	Available	Regional cluster	Aurora PostgreSQL	eu-central-1	2 instances	-	-	-	next window
ga-rageval-dev-bot-app-ehcontextqaragevaldevehcont-9laepxotekbu	Available	Writer instance	Aurora PostgreSQL	eu-central-1a	Serverless v2 (0.5 - 2 ACUs)	35.37%	0 Connections	none	
ga-rageval-dev-bot-app-ehcontextqaragevaldevehcont-ggfrirkfp12w	Available	Reader instance	Aurora PostgreSQL	eu-central-1b	Serverless v2 (0.5 - 2 ACUs)	30.53%	0 Connections	none	

The screenshot shows the 'Modify' dialog for an RDS database. At the top, it says 'Protection turned on'. Below that, 'Database options' show the DB cluster parameter group as 'default.aurora-postgres13'. Under 'Backup', 'Backup retention period' is set to 1 day. Under 'Log exports', 'PostgreSQL log' is selected. Under 'IAM role', 'RDS service-linked role' is chosen. In the 'Maintenance' section, 'Enable auto minor version upgrade' is checked. Under 'DB cluster maintenance window', the start time is set to 00:08 UTC. In the 'Deletion protection' section, the checkbox 'Enable deletion protection' is checked. At the bottom right are 'Cancel' and 'Continue' buttons.

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.

The screenshot shows the AWS S3 Buckets page. The left sidebar includes links for Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens (selected), Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight (7), and AWS Marketplace for S3. The main content area displays an 'Account snapshot' with an 'updated every 24 hours' interval. It shows 18 General purpose buckets. A search bar at the top of the list finds a bucket named 'qa'. The table lists the bucket details:

Name	AWS Region	IAM Access Analyzer	Creation date
qa-rageval-dev-ecs-app-eu-central-1-bucket	Europe (Frankfurt) eu-central-1	View analyzer for eu-central-1	May 31, 2024, 00:13:06 (UTC+02:00)

Actions available for the selected bucket include Copy ARN, Empty, Delete, and Create bucket.

Cleaning up

The image shows two side-by-side terminal windows in the VS Code interface, both titled "aws-genai-rageval-bot".

Left Terminal: Displays the output of a CloudFormation stack deletion command. It shows the deletion of various resources including an AppSync endpoint, an RDS DB instance, and an ECS service. A confirmation message asks if the user wants to delete the "qa-rageval-dev-bot-app". The command used was `cdk destroy --all`.

```
(.venv) + deploy git:(main) ✘ export AWS_PROFILE=blog-profile
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
1:33:29 AM | DELETE_IN_PROGRESS | AWS::ECS::Service
1:33:29 AM | DELETE_IN_PROGRESS | AWS::RDS::DBInstance
1:33:29 AM | DELETE_IN_PROGRESS | AWS::RDS::DBInstance
1:33:30 AM | DELETE_IN_PROGRESS | AWS::CloudFormation::Stack
1:33:31 AM | DELETE_IN_PROGRESS | AWS::CloudFormation::Stack
qa-rageval-dev-bot...D94A3-0MSZU14Q725V
```

Right Terminal: Displays the output of a Docker build process. It shows the creation of a Docker image named "bot-repo" from a Dockerfile located at ".../nc-bot/Dockerfile". The build process involves transferring files from the host to the container, setting up environment variables, and copying application files like requirements_aws.txt and chatbot.py. The final step shows the image being tagged as "bot-repo:qa-app-1.0.0" and pushed to an ECR repository with the ID "602844611342".

```
(.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
  => [internal] load build definition from Dockerfile
  => transferring dockerfile: 919B
  => [internal] load metadata for docker.io/library/python:3.9-slim
  => [internal] load .dockignore
  => transferring context: 2B
  => [1/9] FROM docker.io/library/python:3.9-slim@sha256:088d9217202188598aac37f8db0929345e124a82134ac66b8bb50ee9750b045b
  => [internal] load build context
  => transferring context: 9.10kB
  => CACHED [2/9] RUN apt-get update && apt-get --no-install-recommends install -y build-essential curl software-properties
  => CACHED [3/9] RUN useradd -ms /bin/bash app
  => CACHED [4/9] WORKDIR /home/app
  => CACHED [5/9] COPY requirements_aws.txt /home/app/requirements_aws.txt
  => CACHED [6/9] RUN pip3 install --no-cache-dir -r requirements_aws.txt
  => CACHED [7/9] COPY ./rag_application_framework /home/app/rag_application_framework
  => CACHED [8/9] COPY ./pages /home/app/pages
  => CACHED [9/9] COPY chatbot.py /home/app/chatbot.py
  => exporting to image
  => exporting layers
  => writing image sha256:b5a3dd212f623faaca63e82542a77609e7a993cb5582bbaaa2d70823b2cef743
  => naming to docker.io/library/bot-repo
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
cc9a93b99341: Layer already exists
bf1b154db745: Layer already exists
ae96698df02c: Layer already exists
e555c0055a9b: Layer already exists
205262265e0: Layer already exists
146826fa3ca0: Layer already exists
5d4427064ecc: Layer already exists
qa-app-1.0.0: digest: sha256:054eee2ea625e4912a9c000465225d559173478e5bcb6d331a08357fe4e0ad59 size: 3044
Process completed successfully!
```

Bottom Status Bar: Shows the current workspace status, including the active file "main", other open files like "Launchpad", and sharing options.