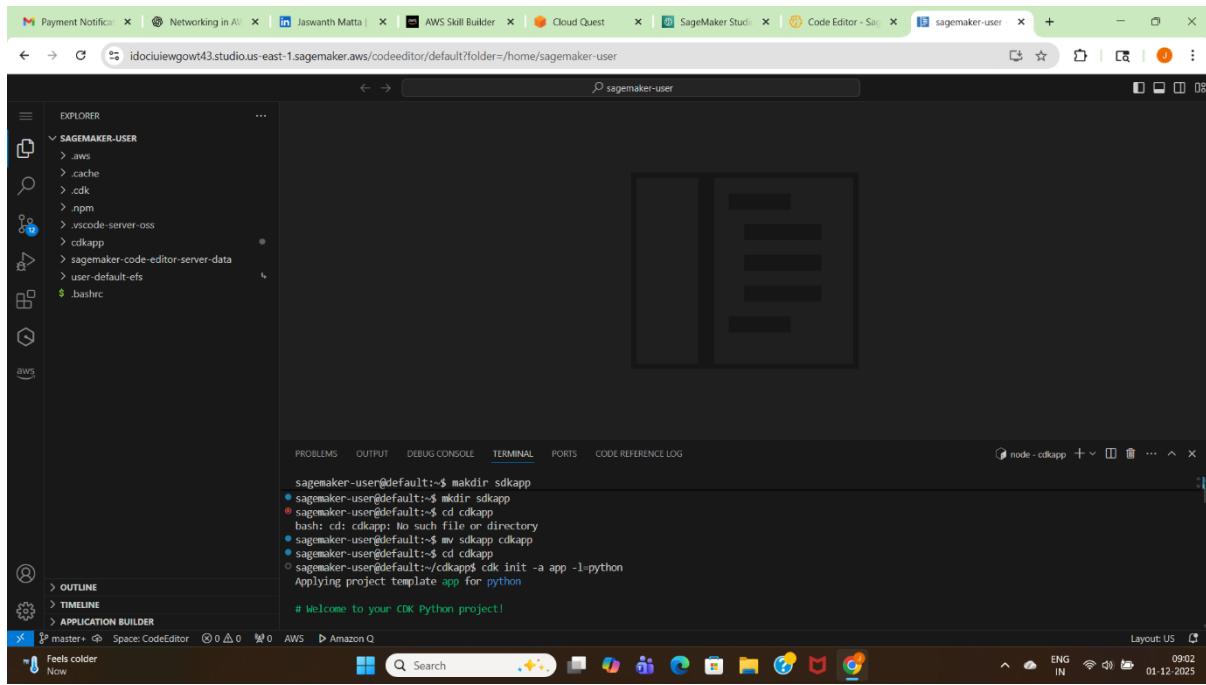
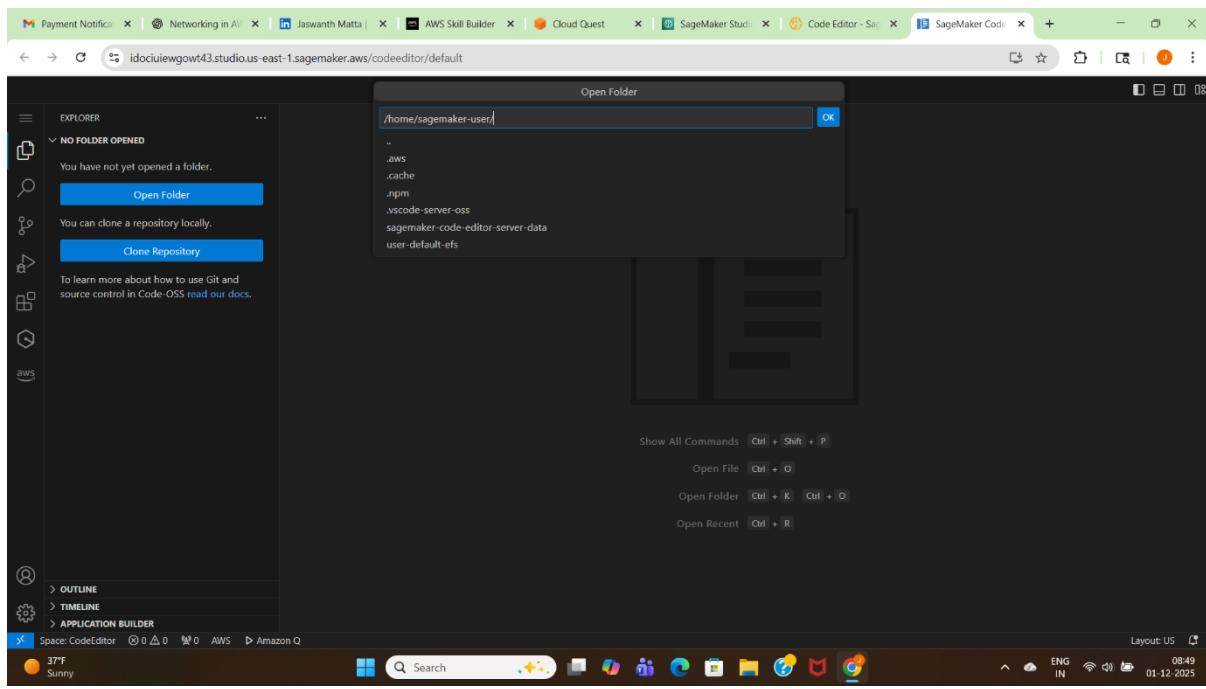


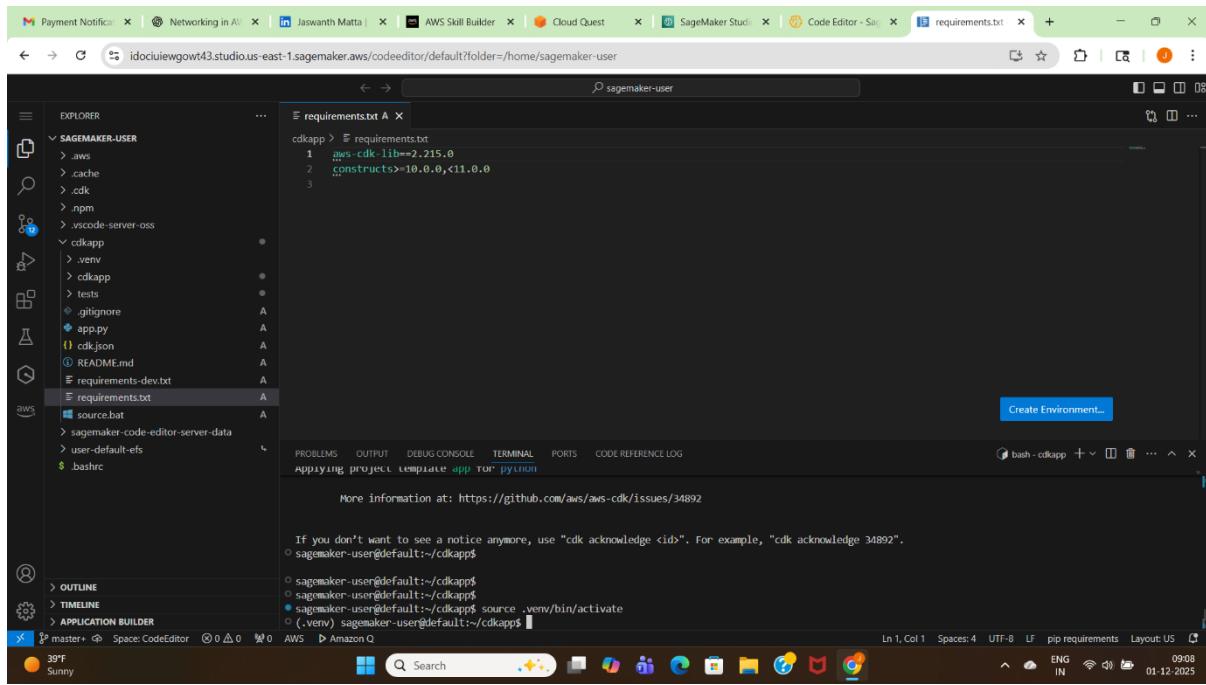
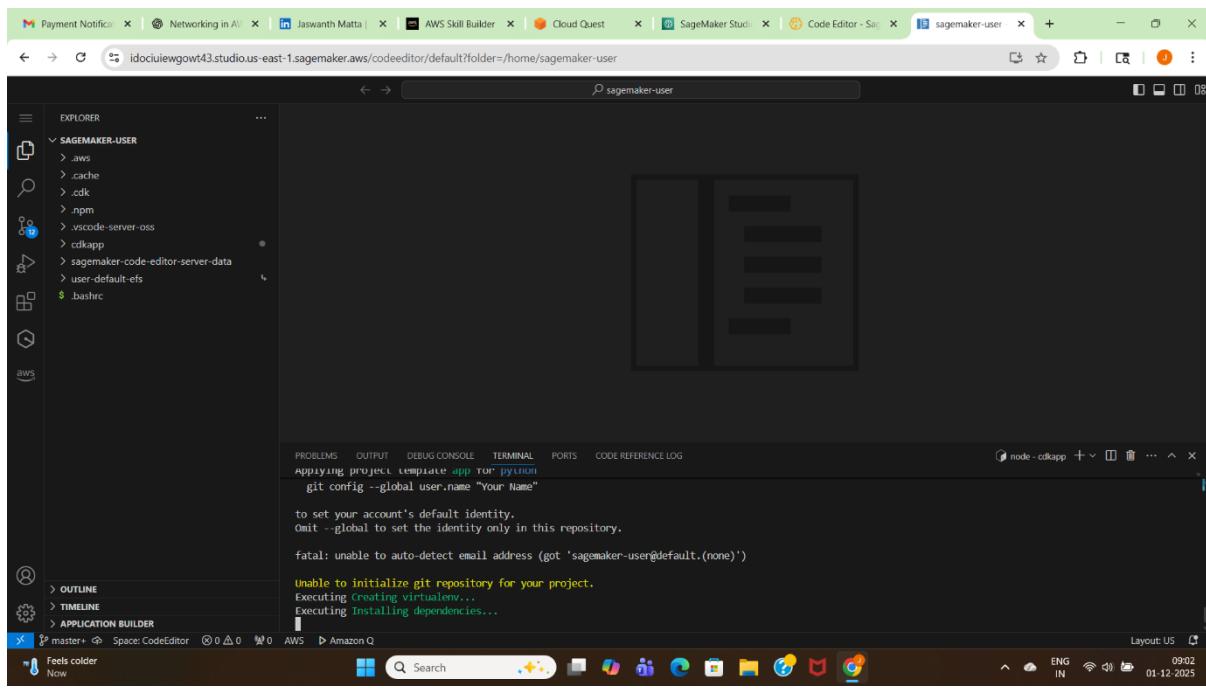
The screenshot shows the Amazon SageMaker AI home page. On the left, there's a sidebar with navigation links like Dashboard, Environment configuration, Applications and IDEs, and SageMaker Studio. The main content area features a large heading "Amazon SageMaker AI" and sub-headings "Build, train, and deploy machine learning models at scale". Below this, there's a section titled "New to SageMaker AI?" with a "Set up for single user" button. Another section titled "Documentation" includes links to Getting started, Tutorials, and Documentation. At the bottom, there's a "SageMaker AI core capabilities" section. The top right of the page shows account information: Account ID: 7349-7944-9836, AWSLabsUser-kBwwJyYtcsTzaPZSc7NYxC/98eaGb..., and the region: United States (N, Virginia). The bottom of the screen shows a Windows taskbar with various icons and system status.

The screenshot shows the SageMaker Studio home page. On the left, there's a sidebar with sections for Applications (JupyterLab, RStudio, Canvas, Code Editor, MLflow), Partner AI Apps (Home), Running instances, Compute, Data, Auto ML, and Experiments. The main content area has a "Home" section with a sub-section "Overview" and a "Getting started" link. It also features cards for "JupyterLab" and "Code Editor". Below this, there's a section titled "Prebuilt and automated solutions" with a "Deploy built-in algorithms, prebuilt solutions, and example notebooks. Build models and prepare data from a visual interface." link. The top right of the page shows account information: Account ID: 7349-7944-9836, AWSLabsUser-kBwwJyYtcsTzaPZSc7NYxC/98eaGb..., and the region: United States (N, Virginia). The bottom of the screen shows a Windows taskbar with various icons and system status.

The screenshot shows the SageMaker Studio Code Editor interface. On the left, there's a sidebar with various application icons like JupyterLab, RStudio, Canvas, Code Editor, and MLflow. Below that is a 'Partner AI Apps' section. The main area displays 'Space templates' with three options: 'Quick start' (ml.t3.medium + 5 GB + 4 GB RAM), 'General purpose CPU' (ml.t3.large + 5 GB + 4 GB RAM), and 'Accelerated compute' (ml.g5.xlarge + 5 GB + 16 GB RAM). A search bar and a filter for 'Running' instances are present. A table lists one instance named 'CodeEditor' which is 'Running' and 'Private'. At the bottom, a banner says 'Introducing spaces' and a message about durable instances.

This screenshot shows the 'Welcome' screen of the SageMaker Studio Code Editor. On the left, there's an 'AMAZON Q CHAT' sidebar with a message from Amazon Q and a link to the AWS Responsible AI Policy. The main area features a 'Get Started with Code Editor' section with a heading 'Discover the best customizations to make Code Editor yours.' It includes a 'Choose your theme' section with four options: 'Dark Modern' (selected), 'Light Modern', 'Dark High Contrast', and 'Light High Contrast'. Below this, there are other customization options: 'Just the right amount of UI', 'Rich support for all your languages', 'Unlock productivity with the Command Palette', and 'Open up your code'. A 'Mark Done' button and a 'Next Section' link are at the bottom. A note about extension auto-update is shown in the bottom right corner.





The screenshot shows the AWS S3 console interface. On the left, the navigation pane includes sections for Buckets, Access management and security, Storage management and insights, Account and organization settings, and AWS Marketplace for S3. The main area displays the contents of the 'lab-bucket-734979449836-125' bucket. The 'Objects' tab is selected, showing three items:

Name	Type	Last modified	Size	Storage class
ci_genai_stack.py	py	December 1, 2025, 08:26:19 (UTC-05:00)	5.9 KB	Standard
userdata.sh	sh	December 1, 2025, 08:26:19 (UTC-05:00)	938.0 B	Standard
vpcapp.zip	zip	December 1, 2025, 08:26:18 (UTC-05:00)	8.4 MB	Standard

A context menu is open over the 'userdata.sh' file, with the 'S3 URI Copied' option highlighted.

The screenshot shows the AWS Code Editor interface running on a SageMaker instance. The terminal window displays the following command being run:

```
Venv Creates a 'venv' virtual environment in the current workspace
```

The terminal also shows the user navigating through files like requirements.txt and source.bat, and executing commands related to the AWS CDK application.

Payment Notifications | Networking in AI | Jaswanth Matta | AWS Skill Builder | Cloud Quest | lab-bucket-7349... | Code Editor - Sag... | userdata.sh - sag... | + | - | X

idociuviewgwt43.studio.us-east-1.sagemaker.aws/codeeditor/default?folder=/home/sagemaker-user

Select an environment type

Venv Creates a 'venv' virtual environment in the current workspace

EXPLORER SAGEMAKER-USER .aws .cache .cdk .npm .vscode-server-oss cdkapp .venv _init_.py cdkapp_stack.py \$ userdata.sh tests .gitignore app.py cdk.json README.md requirements-dev.txt requirements.txt source.bat sagemaker-code-editor-server-data user-default-efs .bashrc

TERMINAL

```
cdkapp > cdkapp
1 #!/bin/bash
2 sudo yum update -y
3 sudo yum -y install jq
4 sudo yum install -y python3
5 sudo dnf install -y pip
6 export bucket_name=$(aws s3api list-buckets --query 'Buckets[*].[Name]' --output text | grep lab-)
7 aws s3 cp s3://$bucket_name/vpcapp.zip .
8 unzip vpcapp.zip
9 export TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" --header "X-aws-ec2-metadata-token-ttl-seconds: 3600")
10 export REGION=$(curl -s http://169.254.169.254/latest/meta-data/placement/region --header "X-aws-ec2-metadata-token: $TOKEN")
11 export DATABASE_HOST=$(aws rds describe-db-clusters --query 'DBClusters[*].[Endpoint]' --output text --region $REGION)
12 export DATABASE_USER=testuser
13 export DATABASE_PASSWORD=password1234!
14 export DATABASE_DB_NAME=population
15 echo $DATABASE_DB_NAME $DATABASE_HOST $DATABASE_PASSWORD $DATABASE_USER $REGION
16 cd vpcapp
17 pip install -r requirements.txt
18 cd loaddatabase
19 python3 database_populate.py
20 cd
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

sagemaker-user@default:~/cdkapp\$

① sagemaker-user@default:~/cdkapp\$ ② sagemaker-user@default:~/cdkapp\$ source .venv/bin/activate
(.venv) sagemaker-user@default:~/cdkapp\$ cd cdkapp
(.venv) sagemaker-user@default:~/cdkapp\$ (.venv) sagemaker-user@default:~/cdkapp\$./cdkapp\$ s3://lab-bucket-734979449836-125/userdata.sh
bash: s3://lab-bucket-734979449836-125/userdata.sh: No such file or directory
(.venv) sagemaker-user@default:~/cdkapp\$./cdkapp\$ aws s3 cp s3://lab-bucket-734979449836-125/userdata.sh .
(.venv) sagemaker-user@default:~/cdkapp\$./cdkapp\$

Ln 1, Col 1 Spaces: 4 UTF-8 LF Shell Script Layout: US

39°F Sunny 01-12-2025

Payment Notifications | Networking in AI | Jaswanth Matta | AWS Skill Builder | Cloud Quest | lab-bucket-7349... | Code Editor - Sag... | cdk.json - sag... | + | - | X

idociuviewgwt43.studio.us-east-1.sagemaker.aws/codeeditor/default?folder=/home/sagemaker-user

Select an environment type

Venv Creates a 'venv' virtual environment in the current workspace

EXPLORER SAGEMAKER-USER .aws .cache .cdk .npm .vscode-server-oss cdkapp .venv _init_.py cdkapp_stack.py \$ userdata.sh tests .gitignore app.py cdk.json README.md requirements-dev.txt requirements.txt source.bat sagemaker-code-editor-server-data user-default-efs .bashrc

TERMINAL

```
cdkapp > cdk
17 "context": {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

sagemaker-user@default:~/cdkapp\$

① sagemaker-user@default:~/cdkapp\$ ② sagemaker-user@default:~/cdkapp\$ source .venv/bin/activate
(.venv) sagemaker-user@default:~/cdkapp\$ cd cdkapp
(.venv) sagemaker-user@default:~/cdkapp\$ (.venv) sagemaker-user@default:~/cdkapp\$./cdkapp\$ s3://lab-bucket-734979449836-125/userdata.sh
bash: s3://lab-bucket-734979449836-125/userdata.sh: No such file or directory
(.venv) sagemaker-user@default:~/cdkapp\$./cdkapp\$ aws s3 cp s3://lab-bucket-734979449836-125/userdata.sh .
(.venv) sagemaker-user@default:~/cdkapp\$./cdkapp\$

Ln 49, Col 59 Spaces: 2 UTF-8 LF {} JSON Layout: US

Feels colder Now 01-12-2025

The screenshot shows the AWS Code Editor interface with a CDK application named "cdkapp". The "EXPLORER" panel on the left lists files and folders: ".aws", ".cache", ".cdk", ".npm", ".vscode-server-oss", "cdkapp", ".venv", "cdkapp", ".init_.py", "cdkapp_stack.py", "# userdata.sh", "tests", ".gitignore", "app.py", "cdk.json", "README.md", "requirements-dev.txt", "requirements.txt", "source.bat", "sagemaker-code-editor-server-data", "user-default-efs", ".bashrc". The "cdkapp_stack.py" file is open in the code editor, showing the following code:

```
from aws_cdk import core, Stack
from aws_cdk import aws_sqs as sqs

class CdkappStack(Stack):
    def __init__(self, scope: construct.Construct, construct_id: str, **kwargs) -> None:
        super().__init__(scope, construct_id, **kwargs)

        # The code that defines your stack goes here

        # example resource
        # queue = sqs.Queue(
        #     self, "CdkappQueue",
        #     visibility_timeout=Duration.seconds(300),
        # )
```

The "CODE REFERENCE LOG" dropdown is open, showing suggestions for "cdkapp > cdkapp". The "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", "PORTS", and "CODE REFERENCE LOG" tabs are visible at the bottom.

The screenshot shows the AWS Code Editor interface with the following details:

- File Explorer:** On the left, it shows the project structure under "SAGEMAKER-USER". The "cdkapp" folder contains files like `__init__.py`, `cdkapp_stack.py`, `userdata.sh`, and `README.md`.
- Code Editor:** The main area displays the `cdkapp_stack.py` file. The code defines a CloudFormation stack named `CdkappStack` that creates a VPC with specific configurations.
- Terminal:** At the bottom, the terminal window shows the command `sageMaker-user@default:~/cdkapp$` followed by the output of running `source .venv/bin/activate` and listing files in the `cdkapp` directory.

A screenshot of a light-colored IDE interface, likely AWS Cloud9 or a similar code editor. The title bar shows multiple tabs including "Payment Notifications", "Networking in AI", "Jaswanth Matta", "AWS Skill Builder", "Cloud Quest", "lab-bucket-7349", "Code Editor - Sag", and "cdkapp_stack.py". The main area has tabs for "requirements.txt", "cdk.json", and "cdkapp_stack.py". The code editor displays the following Python code:

```
class CdkappStack(Construct):
    def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
        super().__init__(scope, construct_id, **kwargs)
        # create a vpc with ipaddresses 10.10.0.0/16, a NAT gateway, a public subnet, PRIVATE_WITH_EGRESS subnet and a RDS subnet
        vpc = ec2.Vpc(
            self,
            "VPC",
            ip_addresses=ec2.IpAddresses.cidr("10.10.0.0/16"),
            subnet_configuration=[
                ec2.SubnetConfiguration(
                    name="Public",
                    subnet_type=ec2.SubnetType.PUBLIC,
                    cidr_mask=24,
                ),
                ec2.SubnetConfiguration(
                    name="PrivateWithEgress",
                    subnet_type=ec2.SubnetType.PRIVATE_WITH_EGRESS,
                    cidr_mask=24,
                ),
                ec2.SubnetConfiguration(
                    name="Private",
                    subnet_type=ec2.SubnetType.PRIVATE_ISOLATED,
                    cidr_mask=24,
                ),
                ec2.SubnetConfiguration(
                    name="RDS",
                    subnet_type=ec2.SubnetType.PRIVATE_ISOLATED,
                    cidr_mask=24,
                )
            ]
        )
```

The left sidebar shows the project structure under "EXPLORER" with files like ".aws", ".cache", ".cdk", ".npm", ".vscode-server-oss", "cdkapp", "cdkapp_stack.py", "userdata.sh", "tests", ".gitignore", "app.py", "cdk.json", "README.md", "requirements-dev.txt", "requirements.txt", "source.bat", "sagemaker-code-editor-server-data", "user-default-efs", and ".bashrc". The bottom status bar shows "master*+", "Space: CodeEditor", "0 △ 0", "AWS", "Amazon Q", "39°F Sunny", "bash - cdkapp", "Ln 43, Col 10", "Spaces: 4", "UTF-8", "LF", "Python 3.12.9 ('base': conda)", "Layout: US", "09:50", "ENG IN", and "01-12-2025".

A screenshot of a dark-themed IDE interface, likely AWS Cloud9 or a similar code editor. The title bar shows the same tabs as the first screenshot. The main area has tabs for "requirements.txt", "cdk.json", and "cdkapp_stack.py". The code editor displays the same Python code as the first screenshot. The left sidebar shows the same project structure. The bottom status bar shows "master*+", "Space: CodeEditor", "0 △ 0", "AWS", "Amazon Q", "39°F Sunny", "bash - cdkapp", "Ln 51, Col 10", "Spaces: 4", "UTF-8", "LF", "Python 3.12.9 ('base': conda)", "Layout: US", "09:53", "ENG IN", and "01-12-2025". A tooltip "Amazon Q Tip 2/3: Invoke suggestions with [Option] + [C] ([Esc] to exit)" is visible at the bottom of the code editor.

The screenshot shows a browser-based code editor interface with multiple tabs open. The active tab is titled "cdkapp_stack.py M". The code in the editor is as follows:

```
12 class CdkappStack(Construct):
13     def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
14         # create a security group for the load balancer
15         alb_security_group = ec2.SecurityGroup(
16             self,
17                 "ALBSecurityGroup",
18                 vpc=vpc,
19                 allow_all_outbound=True,
20         )
21         # create a security group for the RDS instance
22         rds_security_group = ec2.SecurityGroup(
23             self,
24                 "RDSSecurityGroup",
25                 vpc=vpc,
26                 allow_all_outbound=True,
27         )
28         # create a security group for the EC2 instance
29         ec2_security_group = ec2.SecurityGroup(
30             self,
31                 "EC2SecurityGroup",
32                 vpc=vpc,
33                 allow_all_outbound=True,
34         )
35         # add ingress rules for the load balancer security group to allow all traffic on port 80
36         alb_security_group.add_ingress_rule(
37             ec2.Peer.any_ipv4(),
38             ec2.Port.tcp(80),
39         )
40         # add ingress rule for the EC2 instance security group to allow 8443 traffic from the load balancer
41         ec2_security_group.add_ingress_rule(
42             alb_security_group,
43             ec2.Port.tcp(8443),
44         )
```

The code editor has a status bar at the bottom showing "Ln 65, Col 10" and "Python 3.12.9 ('base': conda)".

The screenshot shows the same browser-based code editor interface as the first one, but with additional code added to the end of the file:

```
45         # add ingress rule for the EC2 instance security group to allow 8443 traffic from the load balancer
46         ec2_security_group.add_ingress_rule(
47             alb_security_group,
48             ec2.Port.tcp(8443),
49         )
```

The status bar at the bottom now shows "Ln 75, Col 10" and "Python 3.12.9 ('base': conda)".

The screenshot shows a browser-based code editor interface with multiple tabs open. The active tab is titled "cdkapp_stack.py M". The code in the editor is as follows:

```
12 class CdkappStack(Construct):
13     def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
14         super().__init__(scope, construct_id, **kwargs)
15         vpc=vpc,
16         allow_all_outbound=True,
17     )
18     # add ingress rules for the load balancer security group to allow all traffic on port 80
19     alb_security_group.add_ingress_rule(
20         ec2.Peer.any_ipv4(),
21         ec2.Port.tcp(80),
22     )
23     # add ingress rule for the EC2 instance security group to allow 8443 traffic from the load balancer
24     alb_security_group,
25     ec2.Port.tcp(8443),
26     )
27     # add ingress rule to RDS security group to allow 3306 traffic from EC2 security group
28     rds.security_group.add_ingress_rule(
29         ec2.SecurityGroup,
30         ec2.Port.tcp(3306),
31     )
32     # add ingress rule for the RDS security group to allow 22 from the EC2 instance
33     rds.security_group.add_ingress_rule(
34         ec2.SecurityGroup,
35         ec2.Port.tcp(22),
36     )
```

Below the code editor, the terminal window shows the command: `(.venv) sagemaker-user@default:~/cdkapp/cdkapp$ cd ..`

The status bar at the bottom indicates: "Ln 85, Col 10 Spaces: 4 UTF-8 LF Python 3.12.9 ('base':conda) Layout: US".

The screenshot shows a browser-based code editor interface with multiple tabs open. The active tab is titled "cdkapp_stack.py M". The code in the editor is as follows:

```
12 class CdkappStack(Construct):
13     def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
14         super().__init__(scope, construct_id, **kwargs)
15         # create an rds aurora mysql cluster
16         cluster = rds.DatabaseCluster(self, "MyDatabase",
17             engine = rds.DatabaseClusterEngine.aurora_mysql(version = rds.AuroraMySQLEngineVersion.VER_3_04_0),
18             # credentials using testuser and password1234
19             credentials = rds.Credentials.from_password("testuser", SecretValue.unsafe_plain_text("password1234!")),
20             # add default database name Population
21             default_database_name = "Population",
22             instance_props={
23                 "vpc": vpc,
24                 "security_groups": [rds_sg],
25                 "vpc_subnets": ec2.SubnetSelection(subnet_type = ec2.SubnetType.PRIVATE_ISOLATED)
26             },
27             instances = 1
28         )
29         # define an Amazon Linux 2023 image amzn_linux
30         amzn_linux = ec2.MachineImage.latest_amazon_linux2023(
31             generation=ec2.AmazonLinuxGeneration.AMAZON_LINUX_2023,
32         )
33         # read userdata.sh file from cdkapp directory using readlines
34         userdata = open("cdkapp/userdata.sh", "r").readlines()
35         # Add each line from the script to ec2UserData
36         ec2_user_data = ec2.UserData.for_linux()
```

Below the code editor, the terminal window shows the command: `(.venv) sagemaker-user@default:~/cdkapp/cdkapp$ cd ..`

The status bar at the bottom indicates: "Ln 107, Col 49 Spaces: 4 UTF-8 LF Python 3.12.9 ('base':conda) Layout: US".

```

cdkapp > cdkapp > cdkapp_stack.py > CdkappStack > __init__.py
1 from aws_cdk import (
2     Duration,
3     Stack,
4     SecretValue,
5     aws_ec2 as ec2,
6     aws_rds as rds,
7     aws_iam as iam,
8     aws_elasticloadbalancingv2 as elbv2,
9 )
10 from constructs import Construct
11
12 class Cdkappstack(Stack):
13
14     def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
15         super().__init__(scope, construct_id, **kwargs)
16
17         # create a vpc with IpAddresses 10.10.0.0/16, a NAT gateway, a public subnet, PRIVATE_WITH_EGRESS subnet and a RDS subnet
18         vpc = ec2.Vpc(
19             self,
20             "VPC",
21             ip_addresses=ec2.IpAddresses.cidr("10.10.0.0/16"),
22             subnet_configuration=[
23                 ec2.SubnetConfiguration(
24                     name="Public",
25                     subnet_type=ec2.SubnetType.PUBLIC,
26                     cidr_mask=24,
27                 ),
28                 ec2.SubnetConfiguration(
29                     name="PrivateWithEgress",
30                     subnet_type=ec2.SubnetType.PRIVATE_WITH_EGRESS,
31                     cidr_mask=24,
32                 ),
33             ],
34         )

```

Amazon S3

Buckets

- General purpose buckets
 - Directory buckets
 - Table buckets
 - Vector buckets

Access management and security

- Access Points
- Access Points for FSx
- Access Grants
- IAM Access Analyzer

Storage management and insights

- Storage Lens
- Batch Operations

Account and organization settings

AWS Marketplace for S3

CloudShell Feedback

40°F Sunny

Name	Type	Last modified	Size	Storage class
ci_genial_stack.py	py	December 1, 2025, 08:26:19 (UTC-05:00)	5.9 KB	Standard
userdata.sh	sh	December 1, 2025, 08:26:19 (UTC-05:00)	938.0 B	Standard
vpcapp.zip	zip	December 1, 2025, 08:26:18 (UTC-05:00)	8.4 MB	Standard

The screenshot shows a browser-based code editor interface for AWS CDK. The URL is `lh1r96oqye1ltf8.studio.us-east-1.sagemaker.aws/codeeditor/default?folder=/home/sagemaker-user`. The code editor displays the `cdkapp_stack.py` file, which defines a CloudFront distribution with a Lambda@Edge function and an S3 origin. The code uses the AWS CDK Python library to create resources like VPCs, subnets, and security groups.

```
cdkapp > cdkapp > cdःkapp_stack.py > CdkappStack > __init__  
1  from aws_cdk import (  
2      Duration,  
3      Stack,  
4      SecretValue,  
5      aws_ec2 as ec2,  
6      aws_rds as rds,  
7      aws_iam as iam,  
8      aws_elasticloadbalancingv2 as elbv2,  
9  )  
10 from constructs import Construct  
11  
12 class Cdkappstack(Stack):  
13  
14     def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:  
15         super().__init__(scope, construct_id, **kwargs)  
16  
17     # create a vpc with IpAddresses 10.10.0.0/16, a NAT gateway, a public subnet, PRIVATE_WITH_EGRESS subnet and a RDS subnet  
18     vpc = ec2.Vpc(  
19         self,  
20         "VPC",  
21         ip_addresses=ec2.IpAddresses.cidr("10.10.0.0/16"),  
22         subnet_configuration=[  
23             ec2.SubnetConfiguration(  
24                 name="Public",  
25                 subnet_type=ec2.SubnetType.PUBLIC,  
26                 cidr_mask=24,  
27             ),  
28             ec2.SubnetConfiguration(  
29                 name="PrivateWithEgress",  
30                 subnet_type=ec2.SubnetType.PRIVATE_WITH_EGRESS,  
31                 cidr_mask=24,  
32             ),  
33         ],  
34     )  
35  
36     # create a security group for the load balancer  
37     alb_security_group = ec2.SecurityGroup(  
38         self,  
39         "ALBSecurityGroup",  
40         vpc=vpc,  
41         allow_all_outbound=True,  
42     )  
43  
44     # create a security group for the RDS instance  
45     rds_security_group = ec2.SecurityGroup(  
46         self,  
47         "RDSSecurityGroup",  
48         vpc=vpc,  
49         allow_all_outbound=True,  
50     )  
51  
52     # create a security group for the ECR instance  
53  
54     # create a security group for the Lambda function  
55  
56     # create a security group for the CloudFront distribution  
57  
58     # create a security group for the CloudWatch Metrics  
59  
60     # create a security group for the CloudWatch Metrics  
61  
62     # create a security group for the CloudWatch Metrics  
63  
64     # create a security group for the CloudWatch Metrics  
65  
66     # create a security group for the CloudWatch Metrics  
67  
68     # create a security group for the CloudWatch Metrics  
69  
70     # create a security group for the CloudWatch Metrics  
71  
72     # create a security group for the CloudWatch Metrics  
73  
74     # create a security group for the CloudWatch Metrics  
75  
76     # create a security group for the CloudWatch Metrics  
77  
78     # create a security group for the CloudWatch Metrics  
79  
80     # create a security group for the CloudWatch Metrics  
81  
82     # create a security group for the CloudWatch Metrics  
83  
84     # create a security group for the CloudWatch Metrics  
85  
86     # create a security group for the CloudWatch Metrics  
87  
88     # create a security group for the CloudWatch Metrics  
89  
90     # create a security group for the CloudWatch Metrics  
91  
92     # create a security group for the CloudWatch Metrics  
93  
94  
95     # create a security group for the CloudWatch Metrics  
96  
97     # create a security group for the CloudWatch Metrics  
98  
99     # create a security group for the CloudWatch Metrics  
100    # create a security group for the CloudWatch Metrics  
101  
102    # create a security group for the CloudWatch Metrics  
103  
104    # create a security group for the CloudWatch Metrics  
105  
106    # create a security group for the CloudWatch Metrics  
107  
108    # create a security group for the CloudWatch Metrics  
109  
110    # create a security group for the CloudWatch Metrics  
111  
112    # create a security group for the CloudWatch Metrics  
113  
114    # create a security group for the CloudWatch Metrics  
115  
116    # create a security group for the CloudWatch Metrics  
117  
118    # create a security group for the CloudWatch Metrics  
119  
120    # create a security group for the CloudWatch Metrics  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1749  
1750  

```

```
cdkapp > cdkapp > cdkapp_stack.py > CdappStack > __init__
12     def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
13         super().__init__(scope, construct_id, **kwargs)
14         self.allow_all_outbound = True
15
16         # Create a security group for the EC2 instance
17         ec2_security_group = ec2.SecurityGroup(
18             self,
19             "EC2SecurityGroup",
20             vpc=vpc,
21             allow_all_outbound=True,
22         )
23
24         # Add ingress rules for the load balancer security group to allow all traffic on port 80
25         alb_security_group.add_ingress_rule(
26             ec2.Peer.any_ipv4(),
27             ec2.Port.tcp(80),
28         )
29
30         # Add ingress rule for the EC2 instance security group to allow 8443 traffic from the load balancer
31         ec2_security_group.add_ingress_rule(
32             alb_security_group,
33             ec2.Port.tcp(8443),
34         )
35
36         # Add ingress rule for RDS security group to allow 3306 traffic from EC2 security group
37         rds_security_group.add_ingress_rule(
38             ec2_security_group,
39             ec2.Port.tcp(3306),
40         )
41
42         # Add ingress rule for the RDS security group to allow 22 from the EC2 instance
43         rds_security_group.add_ingress_rule(
44             ec2_security_group,
45             ec2.Port.tcp(22),
46         )
47
48         # Define an RDS Aurora MySQL cluster
49         cluster = rds.DatabaseCluster(self, "MyDatabase",
50             engine=rds.DatabaseClusterEngine.aurora_mysql(version=rds.AuroraMySQLEngineVersion.VER_3_04_0),
51             credentials=rds.Credentials.from_password("testuser", SecretValue.unsafe_plain_text("password1234!")),
52             default_database_name="Population",
53             instance_props={
54                 "vpc": vpc,
55                 "security_groups": [rds_security_group],
56                 "vpc_subnets": ec2.SubnetSelection(subnet_type=ec2.SubnetType.PRIVATE_ISOLATED)
57             },
58             instances=1
59         )
60
61         # Define an Amazon Linux 2023 image amzn_linux
62         amzn_linux = ec2.MachineImage.latest_amazon_linux2023()
63
64         # Read userdata.sh file from cdkapp directory using readline
65         with open("cdkapp/userdata.sh", "r") as f:
66             user_data = f.read()
67
68         # Add each line from the script to ec2 UserData
69         ec2_user_data = ec2.UserData.for_linux()
70         for line in user_data.splitlines():
71             ec2_user_data.append_line(line)
```

```
cdkapp > cdkapp > cdkapp_stack.py > CdappStack > __init__
12     def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
13         super().__init__(scope, construct_id, **kwargs)
14         rds_security_group.add_ingress_rule(
15             ec2_security_group,
16             ec2.Port.tcp(22),
17         )
18
19         # Create an RDS Aurora MySQL cluster
20         cluster = rds.DatabaseCluster(self, "MyDatabase",
21             engine=rds.DatabaseClusterEngine.aurora_mysql(version=rds.AuroraMySQLEngineVersion.VER_3_04_0),
22             credentials=rds.Credentials.from_password("testuser", SecretValue.unsafe_plain_text("password1234!")),
23             default_database_name="Population",
24             instance_props={
25                 "vpc": vpc,
26                 "security_groups": [rds_security_group],
27                 "vpc_subnets": ec2.SubnetSelection(subnet_type=ec2.SubnetType.PRIVATE_ISOLATED)
28             },
29             instances=1
30         )
31
32         # Define an Amazon Linux 2023 image amzn_linux
33         amzn_linux = ec2.MachineImage.latest_amazon_linux2023()
34
35         # Read userdata.sh file from cdkapp directory using readline
36         with open("cdkapp/userdata.sh", "r") as f:
37             user_data = f.read()
38
39         # Add each line from the script to ec2 UserData
40         ec2_user_data = ec2.UserData.for_linux()
41         for line in user_data.splitlines():
42             ec2_user_data.append_line(line)
```

The screenshot shows the AWS Cloud9 IDE interface with the following details:

- Header:** Shows multiple tabs including "Inbox (6,623) - ja", "AWS Skill Builder", "Cloud Quest", "lab-bucket-7349", "Code Editor - S", "Q.Developer not", and "cdkapp_stack.py".
- Left Sidebar (EXPLORER):** Lists the project structure:
 - SAGEMAKER-USER**: .aws, .cache, .cdk, .npm, .vscode-server-oss, cdkapp, .venv, cdk.out, cdkapp, __pycache__, __init__.py, cdkapp_stack.py, userdatas.sh, tests, .gitignore, app.py, cdk.json, README.md, requirements-dev.txt, requirements.txt, source.bat, sagemaker-code-editor-server-data, user-default-efs, bashrc.
 - OUTLINE**
 - TIMELINE**
 - APPLICATION BUILDER**
- Code Editor:** The active file is `cdkapp_stack.py`. The code defines a `CdkappStack` class that constructs an AWS Database instance and associates it with an RDS DB instance. It uses AWS CDK constructs like `Construct`, `ConstructId`, `Resource`, `RDS`, and `DBInstance`.
- Bottom Status Bar:** Displays the current branch as "master", the number of changes (0), the AWS region (US East 1), and the Python version (3.12.9). It also shows the layout configuration as "Layout: US".
- Bottom Icons:** Standard Cloud9 navigation icons for file operations.

(.venv) sagemaker-user@default:/cdkapp\$ cdk synth 2>&1 | tail -n 200

```
[WARNING] aws-cdk-lib/aws.rds.DatabaseClusterProps#instanceProps is deprecated.  
- use writer and readers instead  
This API will be removed in the next major release.  
[WARNING] aws-cdk-lib/aws.rds.DatabaseClusterProps#instances is deprecated.  
- use writer and readers instead  
This API will be removed in the next major release.  
[warning at /cdkappStack/AutoTargetGroup/MyTargetGroup] When creating an empty TargetGroup, you should specify a 'targetType' (this warning may become an error in the future). [ack: @aws-cdk/aws-elbv2.TargetGroup#specifyTargetTypeForEmptyTargetGroup]
```

Resources:

- VPCB9E5F0B4:
 - Type: AWS::EC2::VPC
 - Properties:
 - CidrBlock: 10.10.0.0/16
 - EnableDnsHostnames: true
 - EnableDnsSupport: true
 - InstanceTenancy: default
 - Tags:
 - Key: Name
Value: CdkappStack/VPC
 - Metadata:
 - aws:cdk:path: CdkappStack/VPC/Resource
 - VPCPublicSubnet1SubnetB4246D030:
 - Type: AWS::EC2::Subnet
 - Properties:
 - AvailabilityZone:
 - Fn::Select:
 - Fn::GetAZs: ""
 - CidrBlock: 10.10.0.0/24
 - MapPublicIpOnLaunch: true
 - Tags:
 - Key: aws-cdk:subnet-name
Value: Public
 - Key: aws-cdk:subnet-type

(.venv) sagemaker-user@default:/cdkapp\$ cdk synth

```
Metadata:  
aws:cdk:path: CdkappStack/VPC/PublicSubnet2/RouteTable  
VPCPublicSubnet2RouteTableAssociation5A008732:  
Type: AWS::EC2::SubnetRouteTableAssociation  
Properties:  
RouteTableId:  
Ref: VPCPublicSubnet2RouteTable6F1A15F1  
SubnetId:  
Ref: VPCPublicSubnet2Subnet74179F39  
Metadata:  
aws:cdk:path: CdkappStack/VPC/PublicSubnet2/RouteTableAssociation  
VPCPublicSubnet2DefaultRoute04B1BBA:  
Type: AWS::EC2::Route  
Properties:  
DestinationCidrBlock: 0.0.0.0/0  
GatewayId:  
Ref: VPCVG9998960C  
RouteTableId:  
Ref: VPCPublicSubnet2RouteTable6F1A15F1  
DependOn:  
- VPCVG9998960C  
Metadata:  
aws:cdk:path: CdkappStack/VPC/PublicSubnet2/DefaultRoute  
VPCPublicSubnet2EIP4947BC00:  
Type: AWS::EC2::EIP  
Properties:  
Domain: vpc  
Tags:

- Key: Name  
Value: CdkappStack/VPC/PublicSubnet2

  
Metadata:  
aws:cdk:path: CdkappStack/VPC/PublicSubnet2/EIP  
VPCPublicSubnet2NatGateway3C070193:  
Type: AWS::EC2::NatGateway  
Properties:  
Properties:
```

Inbox (6,623) - j... | LinkedIn (2) Feed | Linked... | AWS Skill Builder | Cloud Quest | lab-bucket-7340 | Code Editor - S... | Q Developer not... | cdkapp_stack.py | + | - | O | X

← → 🔍 lh1r96oqye1ltf8.studio.us-east-1.sagemaker.aws/codeeditor/default?folder=/home/sagemaker-user

JO sageMaker-user

EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

```
(.venv) sageMaker-user@default:~/cdkapp$ cdk synth
  - Fn::Equals:
    - Ref: AWS::Region
    - us-east-1
  - Fn::Equals:
    - Ref: AWS::Region
    - us-east-2
  - Fn::Equals:
    - Ref: AWS::Region
    - us-west-1
  - Fn::Equals:
    - Ref: AWS::Region
    - us-west-2
```

NOTICES (What's this? <https://github.com/aws/aws-cdk/wiki/CLI-Notices>)

34892 CDK CLI will collect telemetry data on command usage starting at version 2.1100.0 (unless opted out)

Overview: We do not collect customer content, and we anonymize the telemetry we do collect. See the attached issue for more information on what data is collected, why, and how to opt-out. Telemetry will NOT be collected for any CDK CLI version prior to version 2.1100.0 - regardless of opt-in/out. You can also preview the telemetry we will start collecting by logging it to a local file, by adding `--unstable-telemetry --telemetry-file=my/local/file` to any `cdk` command.

Affected versions: cli: ^2.0.0

More information at: <https://github.com/aws/aws-cdk/issues/34892>

If you don't want to see a notice anymore, use "cdk acknowledge <id>". For example, "cdk acknowledge 34892".

TIMELINE APPLICATION BUILDER

master+ Space: CodeEditor AWS Amazon Q

Ln 155, Col 9 Spaces: 4 UTF-8 LF Python 3.12.9 (base: conda) Layout: US

36°F Mostly cloudy

Inbox (6,623) - j... | LinkedIn (2) Feed | Linked... | AWS Skill Builder | Cloud Quest | lab-bucket-7340 | Code Editor - S... | Q Developer not... | cdkapp_stack.py | + | - | O | X

← → 🔍 lh1r96oqye1ltf8.studio.us-east-1.sagemaker.aws/codeeditor/default?folder=/home/sagemaker-user

JO sageMaker-user

EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS CODE REFERENCE LOG

```
(.venv) sageMaker-user@default:~/cdkapp$ cdk deploy
[Warning at /dkappstack/ALB/Listener/MyTargetGroup] When creating an empty TargetGroup, you should specify a 'targetType' (this warning may become an error in the future). [ack: @aws-cdk/aws-elbv2:targetGroupSpecifyTargetTypeForEmptyTargetGroup]
```

✖ Synthesis time: 11.5s

current credentials could not be used to assume 'arn:aws:iam::734979449836:role/cdk-hnb059fds-deploy-role-734979449836-us-east-1', but are for the right account. Proceeding anyway.

cdkappStack: start: Building CdkappStack Template

cdkappStack: success: Built CdkappStack Template

cdkappStack: start: Publishing CdkappStack Template (current_account-current_region-6345f3fd)

current credentials could not be used to assume 'arn:aws:iam::734979449836:role/cdk-hnb059fds-file-publishing-role-734979449836-us-east-1', but are for the right account. Proceeding anyway.

cdkappStack: success: Published CdkappStack Template (current_account-current_region-6345f3fd)

current credentials could not be used to assume 'arn:aws:iam::734979449836:role/cdk-hnb059fds-lookup-role-734979449836-us-east-1', but are for the right account. Proceeding anyway.

Lookup role arn:aws:iam::734979449836:role/cdk-hnb059fds-lookup-role-734979449836-us-east-1 was not assumed. Proceeding with default credentials.

Stack CdkappStack
Security Group changes

Group	Dir	Protocol	Peer
+ \${ALBSecurityGroup.GroupId}	In	TCP 80	Everyone (IPv4)
+ \${ALBSecurityGroup.GroupId}	Out	Everything	Everyone (IPv4)
+ \${EC2SecurityGroup.GroupId}	In	TCP 8443	\${ALBSecurityGroup.GroupId}
+ \${EC2SecurityGroup.GroupId}	Out	Everything	Everyone (IPv4)
+ \${IDSecurityGroup.GroupId}	In	TCP 3386	\${EC2SecurityGroup.GroupId}
+ \${IDSecurityGroup.GroupId}	In	TCP 22	\${EC2SecurityGroup.GroupId}
+ \${IDSecurityGroup.GroupId}	Out	Everything	Everyone (IPv4)

(NOTE: There may be security-related changes not in this list. See <https://github.com/aws/aws-cdk/issues/1299>)

--require-approval" is enabled and stack includes security-sensitive updates: 'Do you wish to deploy these changes' (y/n) |

master+ Space: CodeEditor AWS Amazon Q

Ln 155, Col 9 Spaces: 4 UTF-8 LF Python 3.12.9 (base: conda) Layout: US

36°F Mostly cloudy

The screenshot shows the AWS Cloud9 IDE interface with multiple tabs open. The current tab is titled "cdkapp_stack.py". The left sidebar shows a file tree with a "cdkapp" folder containing "cdkapp_stack.py" and other files like ".env", ".gitignore", and "README.md". The main workspace displays the terminal output of a "cdk deploy" command. The terminal output includes:

```
(.venv) sagemaker-user@default:~/cdkapp$ cdk deploy
Stack cdkappStack
Security Group Changes
```

Group	Dir	Protocol	Peer
+\$[ALBSecurityGroup.GroupId]	In	TCP 80	Everyone (IPv4)
+\$[ALBSecurityGroup.GroupId]	Out	Everything	Everyone (IPv4)
+\$[EC2SecurityGroup.GroupId]	In	TCP 8443	\$[ALBSecurityGroup.GroupId]
+\$[EC2SecurityGroup.GroupId]	Out	Everything	Everyone (IPv4)
+\$[RDSecurityGroup.GroupId]	In	TCP 3306	\$[EC2SecurityGroup.GroupId]
+\$[RDSecurityGroup.GroupId]	In	TCP 22	\$[EC2SecurityGroup.GroupId]
+\$[RDSecurityGroup.GroupId]	Out	Everything	Everyone (IPv4)

(NOTE: There may be security-related changes not in this list. See <https://github.com/aws/aws-cdk/issues/1299>)

"--require-approval" is enabled and stack includes security-sensitive updates: 'Do you wish to deploy these changes? (y/n) y

cdkappStack: deploying... [1/1]
current credentials could not be used to assume 'arn:aws:iam::734979449836:role/cdk-hnb69fds-deploy-role-734979449836-us-east-1', but are for the right account. Proceeding anyway.
cdkappStack: creating CloudFormation changeset... [.....] (39/51)

Time	Action	Resource	Stack
5:47:51 AM	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	cdkappStack
5:48:23 AM	CREATE_IN_PROGRESS	AWS::RDS::DBCluster	MyDatabase
5:48:26 AM	CREATE_IN_PROGRESS	AWS::EC2::NatGateway	VPC/PublicSubnet1/NATGateway
5:48:26 AM	CREATE_IN_PROGRESS	AWS::EC2::NatGateway	VPC/PublicSubnet2/NATGateway
5:48:27 AM	CREATE_IN_PROGRESS	AWS::ElasticLoadBalancingV2::LoadBalancer	ALB

At the bottom, the status bar shows "master+" and "Space: CodeEditor".

The screenshot shows a browser window with multiple tabs open, including 'Inbox (6,623)', 'AWS Skill Builder', 'Cloud Quest', 'lab-bucket-7349...', 'Code Editor - S...', 'Q.Developer not...', and 'cdkapp_stack.py'. The main content area displays the AWS CloudFormation console for a stack named 'cdkapp'. The stack summary shows:

- CDK Stack:** cdkappStack
- Deployment Time:** 652.07s
- ARN:** arn:aws:cloudformation:us-east-1:734979449836:stack/cdkappStack/945f3310-d00b-11f0-bfa5-0affff4e1ab5
- Total Time:** 663.57s

The 'NOTICES' section links to <https://github.com/aws/aws-cdk/wiki/CLI-Notices>. A note at the bottom states: "CDK CLI will collect telemetry data on command usage starting at version 2.110.0 (unless opted out). Overview: We do not collect customer content and we anonymize the telemetry we do collect. See the attached issue for more information on what data is collected, why, and how to opt-out. Telemetry will NOT be collected for any CDK CLI version prior to version 2.110.0, regardless of whether you opt-in/out. You can also preview the telemetry we will start collecting by logging it to a local file, by adding '--unstable-telemetry --telemetry-file=/my/local/file' to any 'cdk' command."

```

Inbox (6,623) - ja | LinkedIn (2) Feed | Linkedl | AWS Skill Builder | Cloud Quest | lab-bucket-7340 | Code Editor - S | Q Developer not | cdkapp_stack.py | + - _ X
← → ⌂ lh196oqye1tf8.studio.us-east-1.sagemaker.aws/codeeditor/default?folder=/home/sagemaker-user
bash - cdkapp + ⌂ sagemaker-user
(.venv) sage...@default:~/cdkapp$ cdk deploy

NOTICES (what's this? https://github.com/aws/aws-cdk/wiki/CLI-Notices)
34892 CDK CLI will collect telemetry data on command usage starting at version 2.1100.0 (unless opted out)
Overview: We do not collect customer content and we anonymize the telemetry we do collect. See the attached issue for more information on what data is collected, why, and how to opt-out. Telemetry will NOT be collected for any CDK CLI version prior to version 2.1100.0 - regardless of opt-in/out. You can also preview the telemetry we will start collecting by logging it to a local file, by adding `unstable-telemetry-telemetry-file=my/local/file` to any `cdk` command.

Affected versions: cli: ^2.0.0
More information at: https://github.com/aws/aws-cdk/issues/34892

31885 (cli): Bootstrap stack outdated
Overview: The bootstrap stack in aws://734979449836/us-east-1 is outdated. We recommend at least version 21, distributed with CDK CLI 2.149.0 or higher. Please reboot your environment by running `cdk bootstrap aws://734979449836/us-east-1`.

Affected versions: bootstrap: <v1
More information at: https://github.com/aws/aws-cdk/issues/31885

If you don't want to see a notice anymore, use "cdk acknowledge <id>". For example, "cdk acknowledge 34892".

```

Ln 155, Col 9 Spaces: 4 UTF-8 LF Python 3.12.9 (base: conda) Layout: US

36°F Mostly cloudy

CloudFormation Console now supports multiple new features. See below for more details.

Stacks (5)

Stack name	Status	Created time	Description
CdkappStack	CREATE_COMPLETE	2025-12-03 00:47:40 UTC-0500	This stack includes resources needed to deploy AWS CDK apps into this environment
CDKToolkit	CREATE_COMPLETE	2025-12-02 22:40:18 UTC-0500	This stack includes resources needed to deploy AWS CDK apps into this environment
LabStack-98ea6b70-bd2e-4609-9a08-bd2aa8928a89-jJ2AMTQuTpashH1ARALCPt-1	CREATE_COMPLETE	2025-12-02 22:38:04 UTC-0500	-
LabStack-98ea6b70-bd2e-4609-9a08-bd2aa8928a89-jJ2AMTQuTpashH1ARALCPt-0	CREATE_COMPLETE	2025-12-02 22:38:04 UTC-0500	-
StackSet-AWSLabs-Resources-573773557866-00-0393e17a-421b-4bf4-96db-7ab1f918c79c	UPDATE_COMPLETE	2024-01-30 12:30:58 UTC-0500	-

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

36°F Mostly cloudy

The screenshot shows the AWS CloudFormation console with the 'Resources' tab selected. The table displays the following information:

Logical ID	Type	Status
VPC	-	CREATE_COMPLETE
ALBSecurityGroup	-	CREATE_COMPLETE
RDSSecurityGroup	-	CREATE_COMPLETE
EC2SecurityGroup	-	CREATE_COMPLETE
MyDatabase	-	CREATE_COMPLETE
MyInstance	-	CREATE_COMPLETE
ALB	-	CREATE_COMPLETE
CDKMetadata	-	CREATE_COMPLETE

The screenshot shows the AWS EC2 Load Balancers console for the Cdkapp-ALBAE-xZBHaKByRND4 load balancer. The 'Listeners and rules' tab is selected, showing one rule:

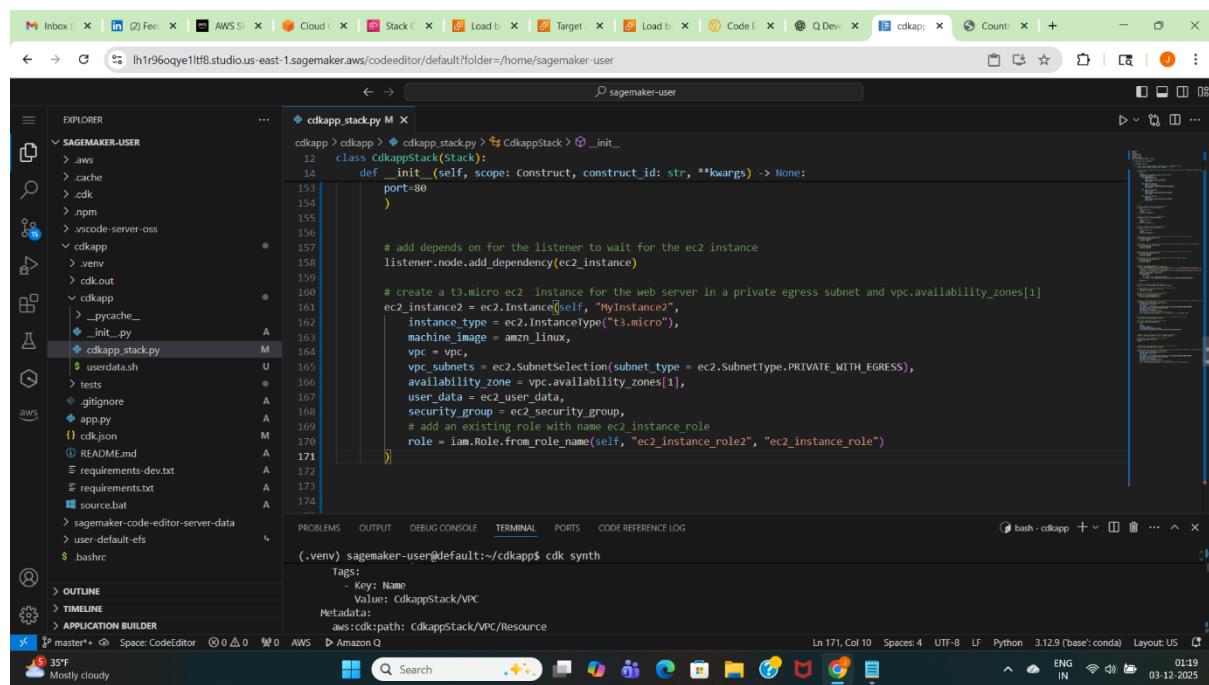
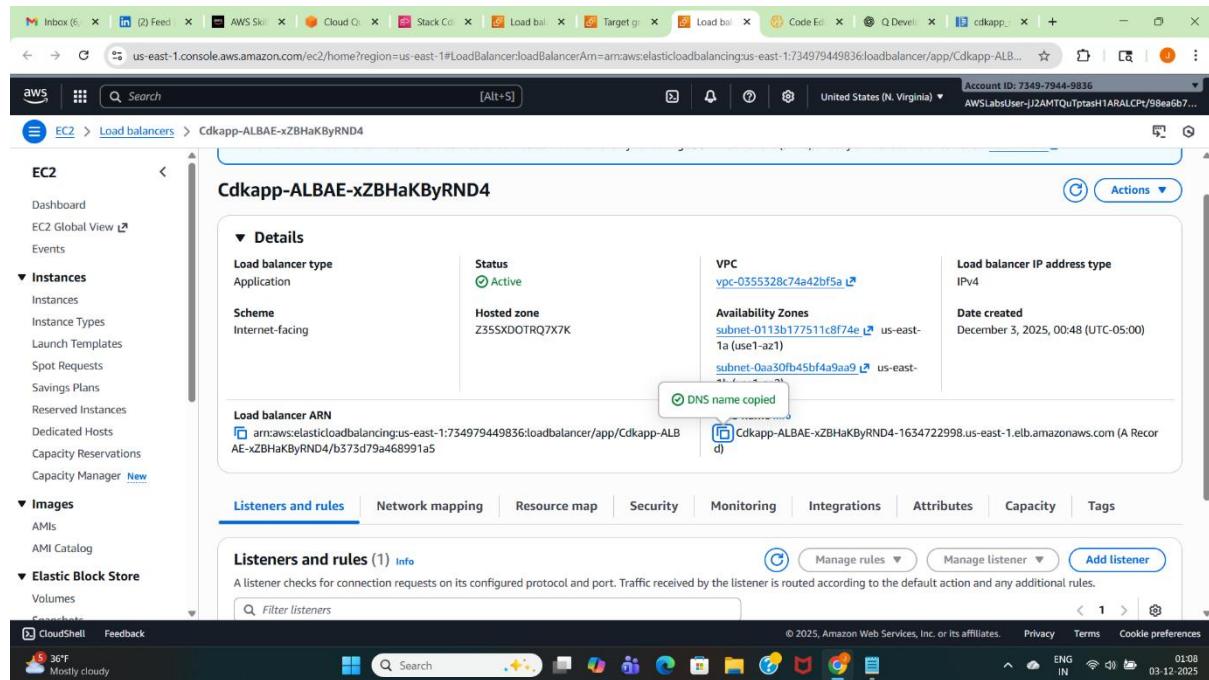
Listener ARN	DNS name info
arn:aws:elasticloadbalancing:us-east-1:734979449836:loadbalancer/app/Cdkapp-ALBAE-xZBHaKByRND4/b373d79a468991a5	Cdkapp-ALBAE-xZBHaKByRND4-1634722998.us-east-1.elb.amazonaws.com (A Record)

The screenshot shows the AWS CloudWatch Metrics Insights interface. At the top, there's a search bar and a navigation bar with tabs like 'Metrics' and 'Logs'. Below the search bar, there's a 'Metrics' section with a table showing metrics from various services. One row is highlighted for 'Amazon CloudWatch Metrics Metrics' with a timestamp of '2023-09-12T12:00:00Z'. The table includes columns for 'Metric Name', 'Unit', 'Value', and 'Dimensions'. A large 'Edit' button is visible at the bottom right of the table.

This screenshot shows the results of a CloudWatch Metrics Insights query. The results are displayed in a table with columns for 'Metric Name', 'Unit', 'Value', and 'Dimensions'. A specific row is highlighted for 'Amazon CloudWatch Metrics Metrics' with a timestamp of '2023-09-12T12:00:00Z'. The table includes columns for 'Metric Name', 'Unit', 'Value', and 'Dimensions'. A large 'Edit' button is visible at the bottom right of the table.

The screenshot shows the AWS CloudWatch Metrics Insights interface. At the top, there's a search bar and a navigation bar with tabs like 'Metrics' and 'Logs'. Below the search bar, there's a 'Metrics' section with a table showing metrics from various services. One row is highlighted, showing 'AWS Lambda' metrics for a specific function. The table includes columns for 'Metric Name', 'Unit', 'Value Type', and 'Last Value'. At the bottom, there's a 'Metrics Insights' section with a table showing log events related to the selected metric.

The screenshot shows the AWS CloudWatch Metrics Insights interface. At the top, there's a search bar and a navigation bar with tabs like 'Metrics' and 'Logs'. Below the search bar, there's a 'Metrics' section with a table showing metrics from various services. One row is highlighted, showing 'AWS Lambda' metrics for a specific function. The table includes columns for 'Metric Name', 'Unit', 'Value Type', and 'Last Value'. At the bottom, there's a 'Metrics Insights' section with a table showing log events related to the selected metric.



The screenshot shows the AWS Cloud9 IDE interface with the following details:

- File Explorer:** On the left, the file structure for the `sagemaker-user` project is displayed. It includes files like `.aws`, `.cache`, `.cdk`, `.npm`, `.vscode-server-oss`, `cdkapp`, `.venv`, `cdk.out`, and `cdkapp` which contains `_pycache_`, `_init_.py`, and `cdkapp_stack.py`.
- Code Editor:** The main editor window shows the `cdkapp_stack.py` file with code related to creating a `CdkappStack`. The terminal output below shows the deployment process.
- Terminal Output:**

```
(.venv) sagemaker-user@default:~/cdkapp$ cdk deploy
cdkappStack: creating cloudformation changeset...
CdkappStack
  Deployment time: 174.82s

Stack ARN:
arn:aws:cloudformation:us-east-1:734974409836:stack/cdkappStack/945f3310-d0b8-11f0-bfa5-0affff4e1ab5

  Total time: 185.51s
```
- NOTICES:** A message from the CDK CLI stating it will collect telemetry data starting at version 2.1100.0.
- Overview:** A note explaining that customer content is anonymized and that telemetry data is collected for research purposes.
- Bottom Status Bar:** Shows the current branch as `master`, the number of changes (0), the AWS region (Amazon Q), the Python version (3.12.9), and the date (03-12-2025).

The screenshot shows the AWS EC2 Target groups page for a specific target group named 'Cdkapp-ALBLI-4X542SB7B82E'. The main summary indicates there are 2 total targets, all of which are healthy. Below this, a table lists the registered targets with their instance IDs, names, ports, zones, health status, and administrative details.

Instance ID	Name	Port	Zone	Health status	Administrative
i-0404577a542d140ea	CdkappStack...	8443	us-east-1b (us...)	Healthy	-
i-057036eb077e7c369	CdkappStack...	8443	us-east-1a (us...)	Healthy	-

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstancesInstanceState=running

EC2 Instances

Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
CdkappStack/...	i-057036eb077e7c369	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1a	-
CdkappStack/...	i-0404577a542d140ea	Running	t3.micro	3/3 checks passed	View alarms +	us-east-1b	-

i-0404577a542d140ea (CdkappStack/MyInstance2)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID i-0404577a542d140ea	Public IPv4 address -	Private IPv4 addresses 10.10.3.205
IPv6 address -	Instance state Running	Public DNS -
Hostname type	Private IP DNS name (IPv4 only)	

CloudShell Feedback

35°F Mostly cloudy

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

ENG IN 01:30 03-12-2025