

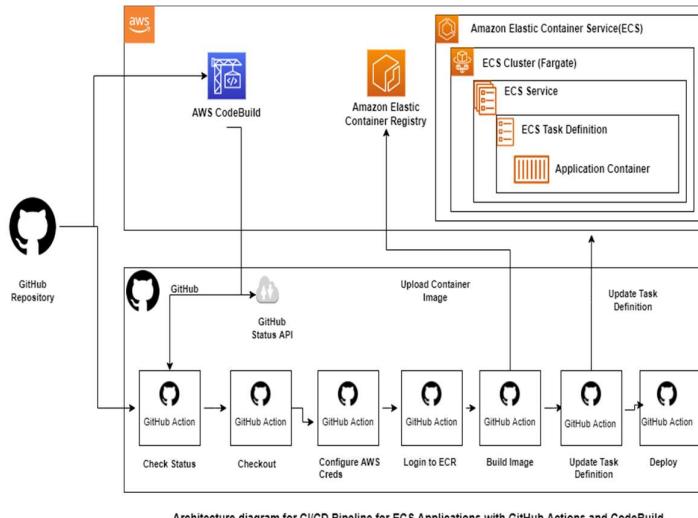
Create a CI/CD pipeline for Amazon ECS with GitHub Actions and AWS CodeBuild Tests

Project Overview: -

Create a CI/CD Pipeline using GitHub actions with GitHub as Source code repository for applications deployed on ECS. This solution also illustrates how AWS CodeBuild can be used with GitHub Actions to execute application tests as part of a complete CI/CD pipeline.

Architecture: -

This architecture represents a complete CI/CD pipeline that uses a GitHub workflow to automatically coordinate building, testing, and deploying an application to ECS for every commit to the repository. This GitHub workflow uses the AWS open-source GitHub Actions to coordinate build and deploy tasks, and uses CodeBuild to execute application tests. We also introduce a custom GitHub Action to this pipeline to evaluate the status of CodeBuild tests.



Architecture diagram for CI/CD Pipeline for ECS Applications with GitHub Actions and CodeBuild

Architecture Implementation	
1	Create a GitHub Repository - GitHub repository consists of a simple web application and accompanying infrastructure files.
2	Create ECS Infrastructure – To build this, we will use AWS Cloud development Kit (CDK). Use CDK to build and deploy a CloudFormation stack that contains the ECS infrastructure required for our application
3	Create CodeBuild Project - CodeBuild is used to execute our application tests and provide the status of these tests to GitHub. webhooks that are automatically created by CodeBuild to trigger a build of our application code every time there is a commit to the master branch of the GitHub repository.

4	Create GitHub Workflow - The goal of workflow is to execute unit tests, build a container image, upload the container image to ECR, and update an ECS Task Definition for every commit to the GitHub repository.
---	--

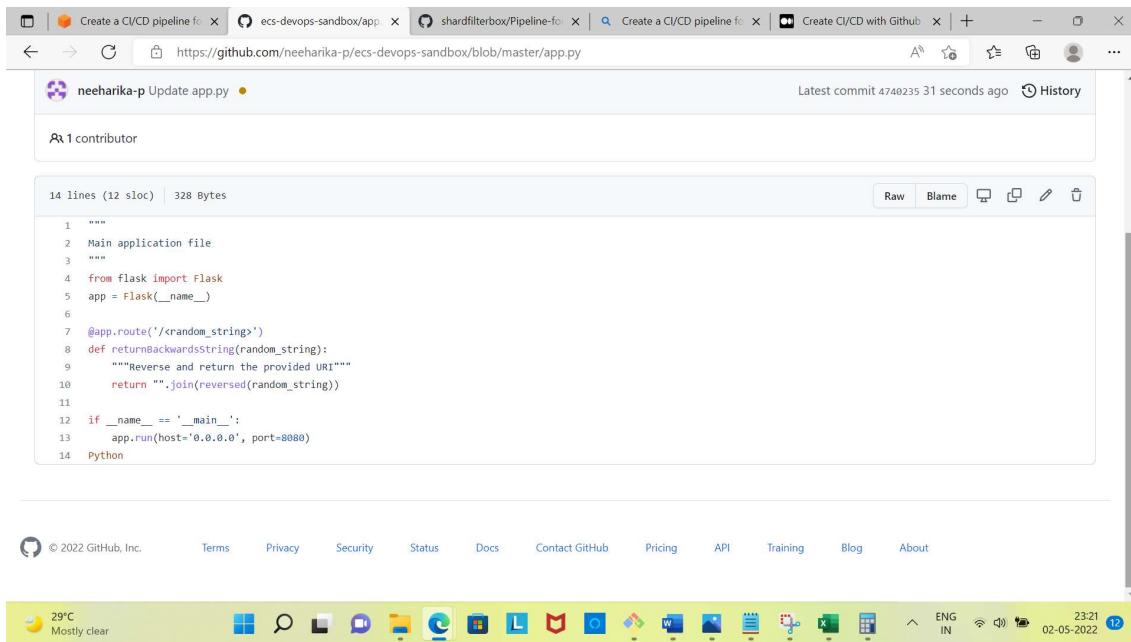
Note: Entire project has been developed using AWS account in us-east-1 region.

Project Implementation – Instructions with Snapshots: -

Step1 : Create GitHub Repository

File	Action	Date
.github/workflows/Update aws.yml	Update aws.yml	14 hours ago
Dockerfile/Update Dockerfile	Update Dockerfile	3 days ago
app.py/Update app.py	Update app.py	13 hours ago
app_test.py/Add files via upload	Add files via upload	3 days ago
buildspec.yml/Add files via upload	Add files via upload	3 days ago
requirements.txt/Add files via upload	Add files via upload	3 days ago
task-definition.json/Add files via upload	Add files via upload	3 days ago

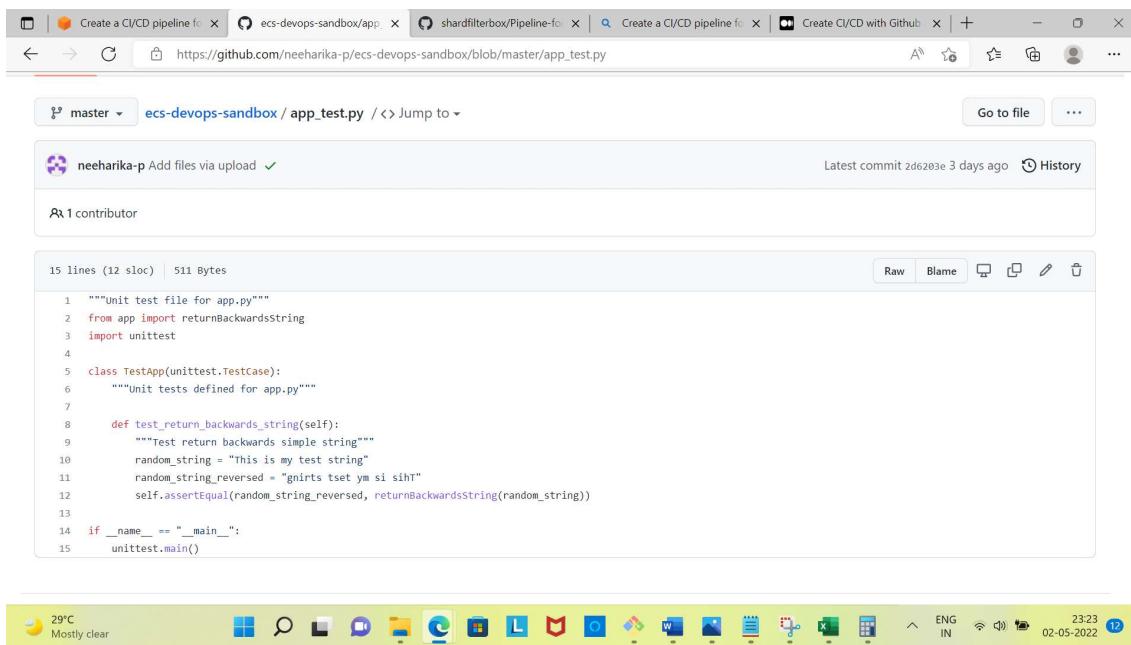
- **app.py:** a simple Flask web application. This application contains one endpoint that reverses and returns the requested URI



A screenshot of a Windows desktop. At the top, there's a browser window showing a GitHub commit for 'app.py' by 'neeharika-p'. The code is a simple Flask application that returns a reversed string. Below the browser is a standard Windows taskbar with icons for File Explorer, Task View, Start, Search, and several pinned applications like Microsoft Edge, File Explorer, and File History.

```
1 """
2 Main application file
3 """
4 from flask import Flask
5 app = Flask(__name__)
6
7 @app.route('/<random_string>')
8 def returnBackwardsString(random_string):
9     """Reverse and return the provided URI"""
10    return ''.join(reversed(random_string))
11
12 if __name__ == '__main__':
13     app.run(host='0.0.0.0', port=8080)
14 Python
```

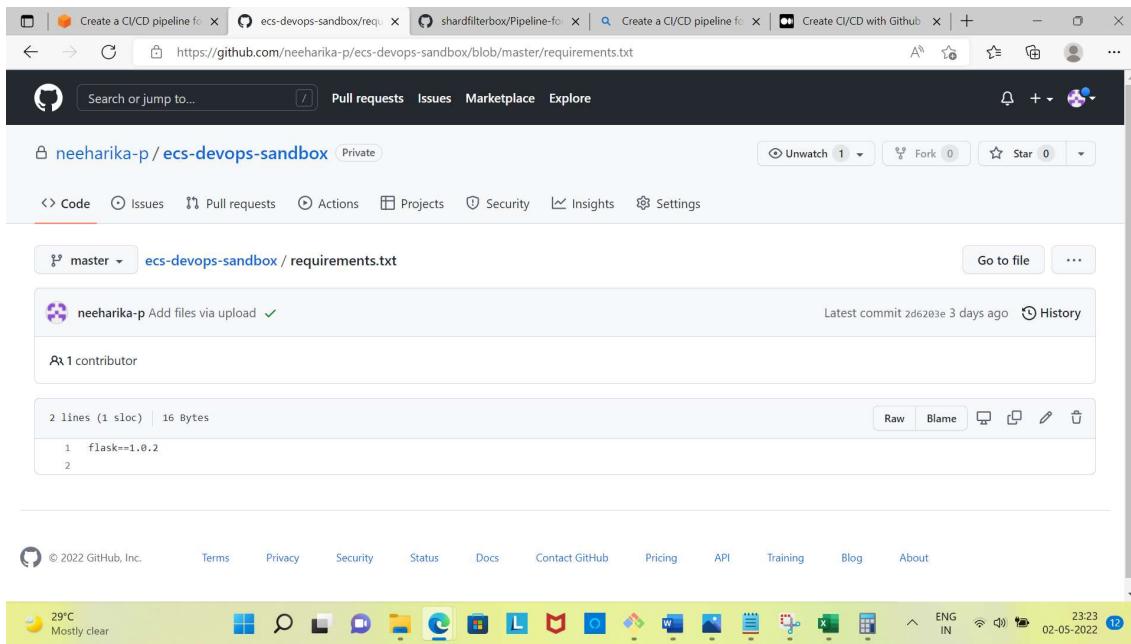
- `app_test.py`: a unit test file for `app.py` to ensure that the string is reversed correctly



A screenshot of a Windows desktop. At the top, there's a browser window showing a GitHub commit for 'app_test.py' by 'neeharika-p'. The code defines a unit test class 'Testapp' that checks if the 'returnBackwardsString' function correctly reverses a string. Below the browser is a standard Windows taskbar with icons for File Explorer, Task View, Start, Search, and several pinned applications like Microsoft Edge, File Explorer, and File History.

```
1 """Unit test file for app.py"""
2 from app import returnBackwardsString
3 import unittest
4
5 class Testapp(unittest.TestCase):
6     """Unit tests defined for app.py"""
7
8     def test_return_backwards_string(self):
9         """Test return backwards simple string"""
10        random_string = "This is my test string"
11        random_string_reversed = "siht si ym tset sihT"
12        self.assertEqual(random_string_reversed, returnBackwardsString(random_string))
13
14 if __name__ == "__main__":
15     unittest.main()
```

- `requirements.txt`: dependencies for `app.py`



<https://github.com/neeharika-p/ecs-devops-sandbox/blob/master/requirements.txt>

neeharika-p / ecs-devops-sandbox · Private

Code Issues Pull requests Actions Projects Security Insights Settings

master · ecs-devops-sandbox / requirements.txt

neeharika-p Add files via upload ✓ Latest commit 2d6203e 3 days ago History

1 contributor

2 lines (1 sloc) | 16 Bytes

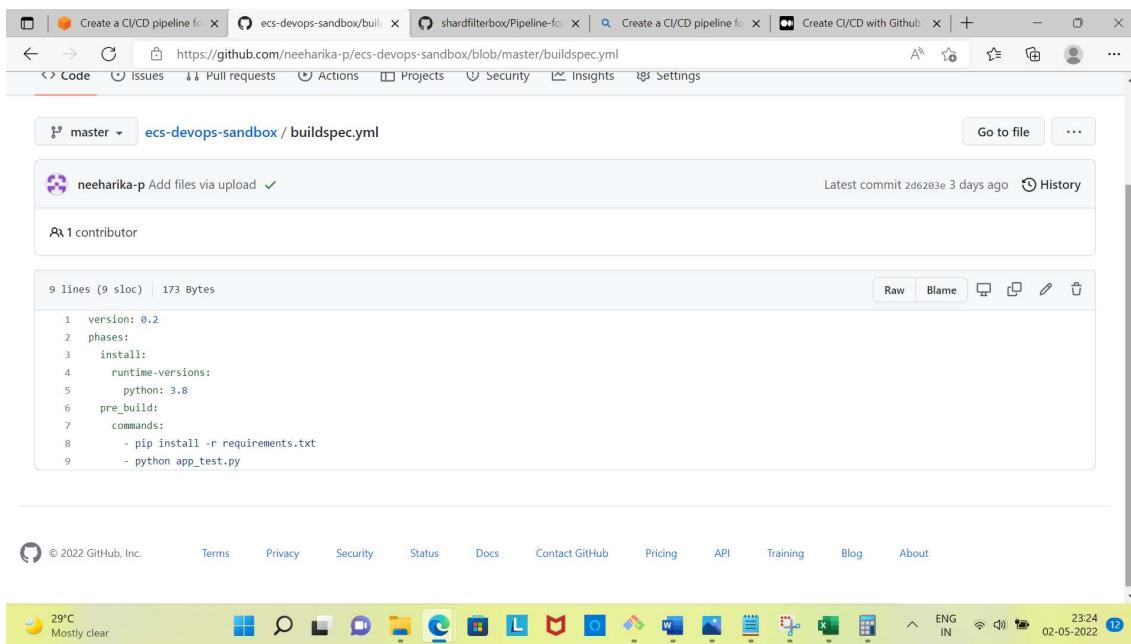
1 flask==1.0.2
2

Raw Blame

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

29°C Mostly clear ENG IN 23:23 02-05-2022 12

- **buildspec.yml: instructions for AWS CodeBuild to run unit tests.**



<https://github.com/neeharika-p/ecs-devops-sandbox/blob/master/buildspec.yml>

neeharika-p Add files via upload ✓ Latest commit 2d6203e 3 days ago History

1 contributor

9 lines (9 sloc) | 173 Bytes

1 version: 0.2
2 phases:
3 install:
4 runtime-versions:
5 python: 3.8
6 pre_build:
7 commands:
8 - pip install -r requirements.txt
9 - python app_test.py

Raw Blame

© 2022 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

29°C Mostly clear ENG IN 23:24 02-05-2022 12

- **Dockerfile: instructions for building the application container image**

A screenshot of a GitHub repository page for 'neeharika-p/ecs-devops-sandbox'. The repository is private. The 'Dockerfile' tab is selected, showing the following code:

```
FROM python:3
RUN python3 -m venv venv
# Set application working directory
WORKDIR /usr/src/app
# Install requirements
COPY requirements.txt .
RUN . /venv/bin/activate & pip install --no-cache-dir -r requirements.txt
# Install application
COPY app.py .
# Run application
CMD python app.py
```

- task-definition.json: specifications for an ECS Task Definition.

A screenshot of a GitHub repository page for 'neeharika-p/ecs-devops-sandbox'. The 'task-definition.json' tab is selected, showing the following JSON code:

```
{ "requiresCompatibilities": [ "FARGATE" ], "inferenceAccelerators": [], "containerDefinitions": [ { "name": "ecs-devops-sandbox", "image": "ecs-devops-sandbox-repository:00000", "resourceRequirements": null, "essential": true, "portMappings": [ { "containerPort": "8888", "protocol": "tcp" } ] }, { "volume": [ { "name": "awslogs", "hostPath": "/var/log/ecs", "mountPoint": "/var/log/ecs" } ] } ], "networkMode": "awsvpc", "memory": "512", "cpu": "256", "executionRoleArn": "arn:aws:iam:089288071887:role/ecs-devops-sandbox-execution-role", "family": "ecs-devops-sandbox-task-definition", "taskRoleArn": "", "placementConstraints": [] }
```

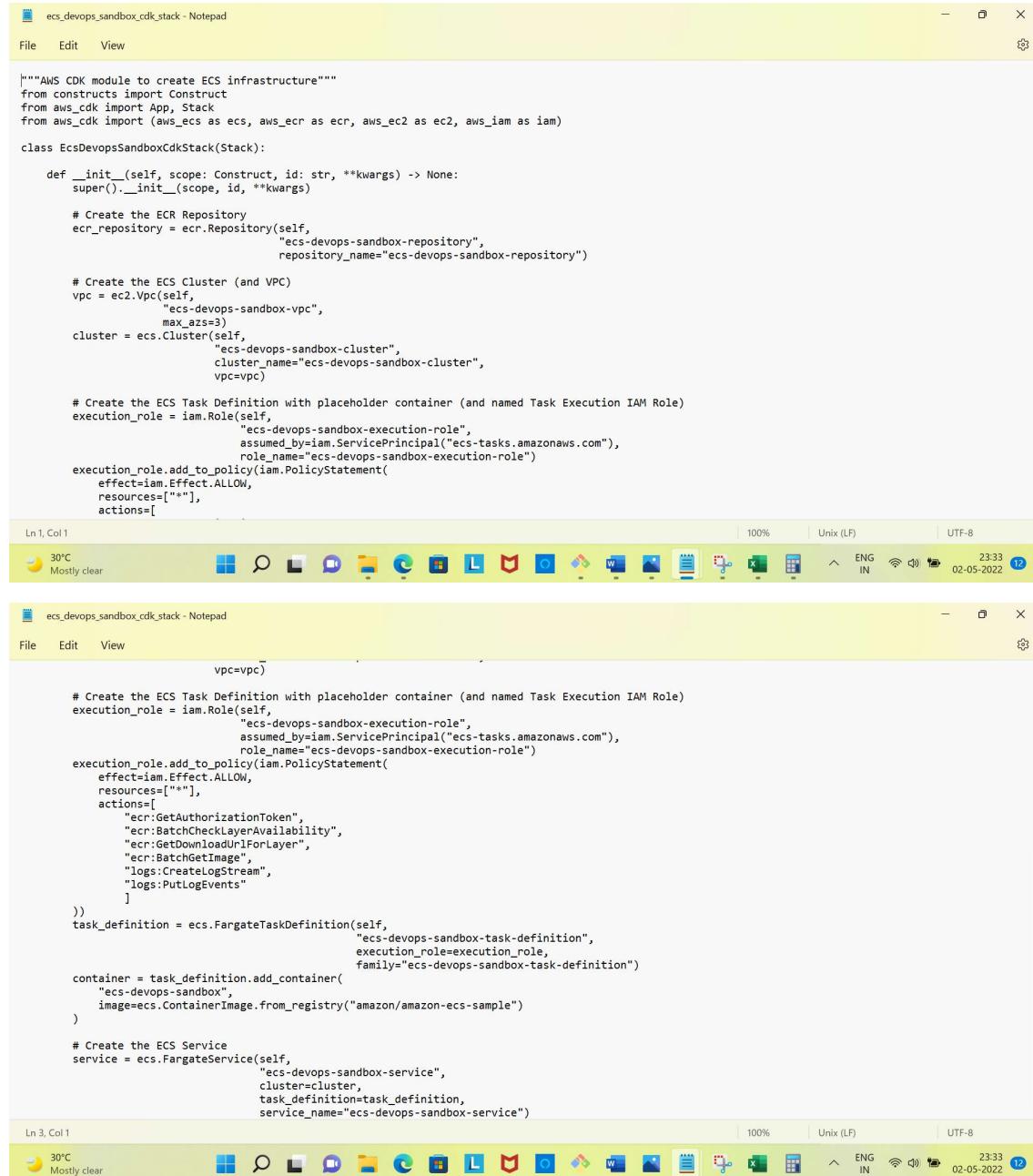
Step2 : Create ECS Infrastructure

- Run following commands to Initialize your CDK Project.

```
# Create a project directory
mkdir ecs-devops-sandbox-cdk
# Enter the directory
cd ecs-devops-sandbox-cdk
# Use the CDK CLI to initiate a Python CDK project
cdk init --language python
```

```
# Activate your Python virtual environment
source .env/Script/Activate.bat
# Install CDK Python general dependencies
pip install -r requirements.txt
# Install CDK Python ECS dependencies
pip install aws_cdk.aws_ec2 aws_cdk.aws_ecs aws_cdk.aws_ecr aws_cdk.aws_iam
```

- Replace contents of ecs_devops_sandbox_cdk/ecs_devops_sandbox_cdk_stack.py



```
ecs_devops_sandbox_cdk_stack - Notepad
File Edit View
"""
AWS CDK module to create ECS infrastructure"""
from constructs import Construct
from aws_cdk import App, Stack
from aws_cdk import (aws_ecs as ecs, aws_ecr as ecr, aws_ec2 as ec2, aws_iam as iam)

class EcsDevopsSandboxCdkStack(Construct):
    def __init__(self, scope: Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

        # Create the ECR Repository
        ecr_repository = ecr.Repository(self,
            "ecs-devops-sandbox-repository",
            repository_name="ecs-devops-sandbox-repository")

        # Create the ECS Cluster (and VPC)
        vpc = ec2.Vpc(self,
            "ecs-devops-sandbox-vpc",
            max_azs=3)
        cluster = ecs.Cluster(self,
            "ecs-devops-sandbox-cluster",
            cluster_name="ecs-devops-sandbox-cluster",
            vpc=vpc)

        # Create the ECS Task Definition with placeholder container (and named Task Execution IAM Role)
        execution_role = iam.Role(self,
            "ecs-devops-sandbox-execution-role",
            assumed_by=iam.ServicePrincipal("ecs-tasks.amazonaws.com"),
            role_name="ecs-devops-sandbox-execution-role")
        execution_role.add_to_policy(iam.PolicyStatement(
            effect=iam.Effect.ALLOW,
            resources=["*"],
            actions=[

Ln 1, Col 1
100% Unix (LF) UTF-8
30°C Mostly clear 23:33 02-05-2022 ENG IN

ecs_devops_sandbox_cdk_stack - Notepad
File Edit View
vpc=vpc)

        # Create the ECS Task Definition with placeholder container (and named Task Execution IAM Role)
        execution_role = iam.Role(self,
            "ecs-devops-sandbox-execution-role",
            assumed_by=iam.ServicePrincipal("ecs-tasks.amazonaws.com"),
            role_name="ecs-devops-sandbox-execution-role")
        execution_role.add_to_policy(iam.PolicyStatement(
            effect=iam.Effect.ALLOW,
            resources=["*"],
            actions=[

                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ]
        ))
        task_definition = ecs.FargateTaskDefinition(self,
            "ecs-devops-sandbox-task-definition",
            execution_role=execution_role,
            family="ecs-devops-sandbox-task-definition")
        container = task_definition.add_container(
            "ecs-devops-sandbox",
            image=ecs.ContainerImage.from_registry("amazon/amazon-ecs-sample")
        )

        # Create the ECS Service
        service = ecs.FargateService(self,
            "ecs-devops-sandbox-service",
            cluster=cluster,
            task_definition=task_definition,
            service_name="ecs-devops-sandbox-service")
Ln 3, Col 1
100% Unix (LF) UTF-8
30°C Mostly clear 23:33 02-05-2022 ENG IN
```

- Run the following command from the root directory of your CDK project. Run cdk bootstrap before cdk deploy for bootstrapping the environment.

```
MINGW64:/c/Users/neeha/Desktop/ecs-devops-sandbox/ecs-devops-sandbox-cdk
```

```
$ cdk bootstrap
  Bootstrapping environment aws://689280071887/us-east-1...
Trusted accounts for deployment: (none)
Trusted accounts for lookup: (none)
Using default execution policy of 'arn:aws:iam::aws:policy/AdministratorAccess'. Pass '--cloudformation-execution-policies' to customize.
CDKToolkit: creating CloudFormation changeset..
CDKToolkit | 0/12 | 8:40:52 pm | REVIEW_IN_PROGRESS | AWS::CloudFormation::Stack | CDKToolkit User Initiated
CDKToolkit | 0/12 | 8:40:52 pm | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | CDKToolkit User Initiated
CDKToolkit | 0/12 | 8:41:07 pm | CREATE_IN_PROGRESS | AWS::ECR::Repository | ContainerAssetsRepository
CDKToolkit | 0/12 | 8:41:07 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | CloudFormationExecutionRole
CDKToolkit | 0/12 | 8:41:07 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | LookupRole
CDKToolkit | 0/12 | 8:41:08 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | FilePublishingRole
CDKToolkit | 0/12 | 8:41:08 pm | CREATE_IN_PROGRESS | AWS::S3::Bucket | StagingBucket
CDKToolkit | 0/12 | 8:41:08 pm | CREATE_IN_PROGRESS | AWS::SSM::Parameter | CdkBootstrapVersion
CDKToolkit | 0/12 | 8:41:08 pm | CREATE_IN_PROGRESS | AWS::SSM::Parameter | ImagePublishingRole
CDKToolkit | 0/12 | 8:41:08 pm | CREATE_IN_PROGRESS | AWS::SSM::Parameter | CloudFormationExecutionRole Resource creation Initiated
CDKToolkit | 0/12 | 8:41:08 pm | CREATE_IN_PROGRESS | AWS::SSM::Parameter | FilePublishingRole Resource creation Initiated
CDKToolkit | 0/12 | 8:41:09 pm | CREATE_IN_PROGRESS | AWS::SSM::Parameter | ImagePublishingRole Resource creation Initiated
CDKToolkit | 0/12 | 8:41:09 pm | CREATE_IN_PROGRESS | AWS::S3::Bucket | StagingBucket Resource creation Initiated
CDKToolkit | 0/12 | 8:41:09 pm | CREATE_IN_PROGRESS | AWS::S3::Bucket | LookAndFeelResource creation Initiated
CDKToolkit | 0/12 | 8:41:09 pm | CREATE_IN_PROGRESS | AWS::S3::Bucket | ContainerAssetsRepository Resource creation Initiated
CDKToolkit | 0/12 | 8:41:10 pm | CREATE_COMPLETE | AWS::ECR::Repository | ContainerAssetsRepository
CDKToolkit | 1/12 | 8:41:10 pm | CREATE_IN_PROGRESS | AWS::SSM::Parameter | CdkBootstrapVersion Resource creation Initiated
CDKToolkit | 2/12 | 8:41:12 pm | CREATE_COMPLETE | AWS::SSM::Parameter | CdkBootstrapVersion
CDKToolkit | 3/12 | 8:41:23 pm | CREATE_COMPLETE | AWS::IAM::Role | CloudFormationExecutionRole
CDKToolkit | 4/12 | 8:41:23 pm | CREATE_COMPLETE | AWS::IAM::Role | FilePublishingRole
CDKToolkit | 5/12 | 8:41:23 pm | CREATE_COMPLETE | AWS::IAM::Role | ImagePublishingRole
CDKToolkit | 6/12 | 8:41:23 pm | CREATE_COMPLETE | AWS::IAM::Role | LookupRole
CDKToolkit | 6/12 | 8:41:25 pm | CREATE_IN_PROGRESS | AWS::IAM::Policy | ImagePublishingRoleDefaultPolicy
CDKToolkit | 6/12 | 8:41:26 pm | CREATE_IN_PROGRESS | AWS::IAM::Policy | ImagePublishingRoleDefaultPolicy Resource creation Initiated
CDKToolkit | 7/12 | 8:41:30 pm | CREATE_COMPLETE | AWS::S3::Bucket | StagingBucket
CDKToolkit | 7/12 | 8:41:33 pm | CREATE_IN_PROGRESS | AWS::IAM::Policy | FilePublishingRoleDefaultPolicy
CDKToolkit | 7/12 | 8:41:33 pm | CREATE_IN_PROGRESS | AWS::SSM::Parameter | StagingBucketPolicy
CDKToolkit | 7/12 | 8:41:33 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | DeploymentActionRole
CDKToolkit | 7/12 | 8:41:33 pm | CREATE_IN_PROGRESS | AWS::IAM::Policy | FilePublishingRoleDefaultPolicy Resource creation Initiated
CDKToolkit | 7/12 | 8:41:34 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | DeploymentActionRole Resource creation Initiated
CDKToolkit | 7/12 | 8:41:34 pm | CREATE_IN_PROGRESS | AWS::S3::BucketPolicy | StagingBucketPolicy Resource creation Initiated
CDKToolkit | 8/12 | 8:41:34 pm | CREATE_COMPLETE | AWS::S3::BucketPolicy | StagingBucketPolicy
CDKToolkit | 10/12 | 8:41:46 pm | CREATE_COMPLETE | AWS::IAM::Policy | FilePublishingRoleDefaultPolicy
CDKToolkit | 11/12 | 8:41:48 pm | CREATE_COMPLETE | AWS::IAM::Role | DeploymentActionRole
CDKToolkit | 12/12 | 8:41:52 pm | CREATE_COMPLETE | AWS::CloudFormation::Stack | CDKToolkit
 Environment aws://689280071887/us-east-1 bootstrapped.
```

```
MINGW64:/c/Users/neeha/Desktop/ecs-devops-sandbox/ecs-devops-sandbox-cdk
```

```
$ cdk deploy
* Synthesis time: 11.01s

EcsDevopsSandboxCdkStack: deploying...
[0%] start: Publishing 8def1b4fab29d380ff774dc6b598fa463c39f99adf303d357bbc35190b277:current_account-current_region
EcsDevopsSandboxCdkStack: creating CloudFormation changeset...
EcsDevopsSandboxCdkStack | 0/33 | 8:43:10 pm | REVIEW_IN_PROGRESS | AWS::CloudFormation::Stack | EcsDevopsSandboxCdkStack User Initiated
EcsDevopsSandboxCdkStack | 0/33 | 8:43:21 pm | CREATE_IN_PROGRESS | AWS::CloudFormation::Stack | EcsDevopsSandboxCdkStack User Initiated
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::CDK::Metadata | CDKMetadata/Default (CDKMetadata)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::EC2::VPC | ecs-devops-sandbox-vpc (ecsdevopssandboxvpc7A58AE2A)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::ECR::Repository | ecs-devops-sandbox-repository (ecsdevopssandboxrepository8013CS1)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | ecs-devops-sandbox-task-definition/TaskRole (ecsdevopssandboxtaskdefinition)
taskdefinitionTaskRole@09087683E)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::EC2::Cluster | ecs-devops-sandbox-cluster (ecsdevopssandboxcluster26C7A362)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | ecs-devops-sandbox-execution-role (ecsdevopssandboxexecution)
01ef3e89161) Resource creation Initiated
PublicSubnet1ETP603987C3)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::EC2::InternetGateway | ecs-devops-sandbox-vpc/IGW (ecsdevopssandboxvpcIGW866A0184)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | ecs-devops-sandbox-task-definition/TaskRole (ecsdevopssandboxtaskdefinition)
taskdefinitionTaskRole@09087683E) Resource creation Initiated
EcsDevopsSandboxCdkStack | 0/33 | 8:43:28 pm | CREATE_IN_PROGRESS | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet2/EIP (ecsdevopssandboxvpcPublicSubnet2)
EcsDevopsSandboxCdkStack | 0/33 | 8:43:29 pm | CREATE_IN_PROGRESS | AWS::EC2::VPC | ecs-devops-sandbox-vpc (ecsdevopssandboxvpc7A58AE2A) Resource
creation Initiated
EcsDevopsSandboxCdkStack | 0/33 | 8:43:29 pm | CREATE_IN_PROGRESS | AWS::IAM::Role | ecs-devops-sandbox-execution-role (ecsdevopssandboxexecution)
01ef3e89161) Resource creation Initiated
EcsDevopsSandboxCdkStack | 0/33 | 8:43:29 pm | CREATE_IN_PROGRESS | AWS::CDK::Metadata | CDKMetadata/Default (CDKMetadata) Resource creation Initiated
EcsDevopsSandboxCdkStack | 1/33 | 8:43:30 pm | CREATE_COMPLETE | AWS::CDK::Metadata | CDKMetadata/Default (CDKMetadata)
EcsDevopsSandboxCdkStack | 1/33 | 8:43:30 pm | CREATE_IN_PROGRESS | AWS::EC2::InternetGateway | ecs-devops-sandbox-vpc/IGW (ecsdevopssandboxvpcIGW866A0184) R
esource creation Initiated
EcsDevopsSandboxCdkStack | 1/33 | 8:43:30 pm | CREATE_IN_PROGRESS | AWS::ECR::Repository | ecs-devops-sandbox-repository (ecsdevopssandboxrepository8013CS1)
EcsDevopsSandboxCdkStack | 1/33 | 8:43:31 pm | CREATE_COMPLETE | AWS::ECR::Repository | ecs-devops-sandbox-repository (ecsdevopssandboxrepository8013CS1)
EcsDevopsSandboxCdkStack | 2/33 | 8:43:31 pm | CREATE_IN_PROGRESS | AWS::EC2::Cluster | ecs-devops-sandbox-cluster (ecsdevopssandboxcluster26C7A362)
Resource creation Initiated
EcsDevopsSandboxCdkStack | 3/33 | 8:43:35 pm | CREATE_COMPLETE | AWS::EC2::Cluster | ecs-devops-sandbox-cluster (ecsdevopssandboxcluster26C7A362)
EcsDevopsSandboxCdkStack | 4/33 | 8:43:43 pm | CREATE_COMPLETE | AWS::IAM::Role | ecs-devops-sandbox-task-definition/TaskRole (ecsdevopssandboxtaskdefinition)
taskdefinitionTaskRole@09087683E)
EcsDevopsSandboxCdkStack | 5/33 | 8:43:43 pm | CREATE_COMPLETE | AWS::IAM::Role | ecs-devops-sandbox-execution-role (ecsdevopssandboxexecution)
01ef3e89161)
```

```
MINGW64/c/Users/neeha/Desktop/ecs-devops-sandbox/ecs-devops-sandbox-cdk taskdefnitionTaskRole19d87689E | AWS::IAM::Role | ecs-devops-sandbox-execution-role (ecsdevopssandboxexecutionrole)
taskdefnitionTaskRole19d87689E | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet1/EIP (ecsdevopssandboxvpc)
oldEB9161 | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet2/EIP (ecsdevopssandboxvpc)
EcsDevopsSandboxCdkStack | / 5/33 | 8:43:43 pm | CREATE_IN_PROGRESS | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet1/EIP (ecsdevopssandboxvpc)
PublicSubnet1EIP03987C3 | Resource creation Initiated | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet2/EIP (ecsdevopssandboxvpc)
EcsDevopsSandboxCdkStack | / 5/33 | 8:43:43 pm | CREATE_IN_PROGRESS | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet1/EIP (ecsdevopssandboxvpc)
PublicSubnet2EIPB7386941 | Resource creation Initiated | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet2/EIP (ecsdevopssandboxvpc)
EcsDevopsSandboxCdkStack | / 6/33 | 8:43:44 pm | CREATE_COMPLETE | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet1/EIP (ecsdevopssandboxvpc)
EcsDevopsSandboxCdkStack | / 7/33 | 8:43:45 pm | CREATE_COMPLETE | AWS::EC2::EIP | ecs-devops-sandbox-vpc/PublicSubnet2/EIP (ecsdevopssandboxvpc)
EcsDevopsSandboxCdkStack | / 7/33 | 8:43:45 pm | CREATE_IN_PROGRESS | AWS::IAM::Policy | ecs-devops-sandbox-execution-role/DefaultPolicy(48CD024F) | ecs-devops-sandbox-task-definition (ecsdevopssandboxtaskdefinition)
EcsDevopsSandboxCdkStack | / 7/33 | 8:43:46 pm | CREATE_IN_PROGRESS | AWS::ECS::TaskDefinition | ecs-devops-sandbox-vpc/(ecsdevopssandboxvpc7A58AE2A) | ecs-devops-sandbox-execution-role/DefaultPolicy(48CD024F) | ecs-devops-sandbox-task-definition (ecsdevopssandboxtaskdefinition)
EcsDevopsSandboxCdkStack | / 8/33 | 8:43:46 pm | CREATE_COMPLETE | AWS::EC2::VPC | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 8/33 | 8:43:46 pm | CREATE_IN_PROGRESS | AWS::IAM::Policy | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:47 pm | CREATE_COMPLETE | AWS::EC2::InternetGateway | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:48 pm | CREATE_IN_PROGRESS | AWS::EC2::RouteTable | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:48 pm | CREATE_IN_PROGRESS | AWS::EC2::RouteTable | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:48 pm | CREATE_IN_PROGRESS | AWS::EC2::SecurityGroup | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:48 pm | CREATE_IN_PROGRESS | AWS::EC2::RouteTable | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:48 pm | CREATE_IN_PROGRESS | AWS::ECS::TaskDefinition | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-task-definition (ecsdevopssandboxtaskdefinition)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::Subnet | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::Subnet | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::Subnet | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::ECS::TaskDefinition | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-task-definition (ecsdevopssandboxtaskdefinition)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::Subnet | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::RouteTable | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::RouteTable | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 9/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::VPCGatewayAttachment | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/VPCGW (ecsdevopssandboxvpcvPCGW83A748)
EcsDevopsSandboxCdkStack | / 10/33 | 8:43:49 pm | CREATE_COMPLETE | AWS::ECS::TaskDefinition | ecs-devops-sandbox-task-definition (ecsdevopssandboxtaskdefinition)
EcsDevopsSandboxCdkStack | / 10/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::VPCGatewayAttachment | ecs-devops-sandbox-vpc/VPCGW (ecsdevopssandboxvpcvPCGW83A748)
EcsDevopsSandboxCdkStack | / 10/33 | 8:43:49 pm | CREATE_IN_PROGRESS | AWS::EC2::RouteTable | ecs-devops-sandbox-vpc/PublicSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PublicSubnet1RouteTable(e1420)
EcsDevopsSandboxCdkStack | / 10/33 | 8:43:50 pm | CREATE_IN_PROGRESS | AWS::EC2::Subnet | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420) | ecs-devops-sandbox-vpc/PrivateSubnet1RouteTable(e1420)
```

```
MINGW64\c/Users/neeha/Desktop/ecs-devops-sandbox/ecs-devops-sandbox-cdk

ecs-devopsSandboxCdkStack | 23/33 | 8:44:08 pm | CREATE_IN_PROGRESS | AWS:EC2::RouteTableAssociation | ecs-devops-sandbox-vpc/PublicSubnet2/DefaultRoute (ecsdevops
ecs-devopsSandboxCdkStack | 23/33 | 8:44:08 pm | CREATE_IN_PROGRESS | AWS:EC2::RouteTableAssociation | ecs-devops-sandbox-vpc/PublicSubnet1/DefaultRoute (ecsdevops
andboxvpcPublicSubnet1DefaultRoute03AB353) Resource creation Initiated
EcsDevopsSandboxCdkStack | 23/33 | 8:44:08 pm | CREATE_COMPLETE | AWS:EC2::SubnetRouteTableAssociation | ecs-devops-sandbox-vpc/PrivateSubnet1/RouteTableAssociation (e
EcsDevopsSandboxCdkStack | 23/33 | 8:44:08 pm | CREATE_COMPLETE | AWS:EC2::SubnetRouteTableAssociation | ecs-devops-sandbox-vpc/PublicSubnet1/RouteTableAssociation (e
EcsDevopsSandboxCdkStack | 23/33 | 8:44:08 pm | CREATE_IN_PROGRESS | AWS:EC2::RouteTableAssociation | ecs-devops-sandbox-vpc/PrivateSubnet2/RouteTableAssociation (e
EcsDevopsSandboxCdkStack | 23/33 | 8:44:08 pm | CREATE_IN_PROGRESS | AWS:EC2::SubnetRouteTableAssociation | ecs-devops-sandbox-vpc/PrivateSubnet2/RouteTableAssociation (e
EcsDevopsSandboxCdkStack | 23/33 | 8:44:08 pm | CREATE_COMPLETE | AWS:EC2::SubnetRouteTableAssociation | ecs-devops-sandbox-vpc/PublicSubnet1/RouteTableAssociation (e
EcsDevopsSandboxCdkStack | 24/33 | 8:44:08 pm | CREATE_COMPLETE | AWS:EC2::SubnetRouteTableAssociation | ecs-devops-sandbox-vpc/PrivateSubnet2/RouteTableAssociation (e
EcsDevopsSandboxCdkStack | 25/33 | 8:44:08 pm | CREATE_COMPLETE | AWS:EC2::RouteTableAssociation | ecs-devops-sandbox-vpc/PublicSubnet2/DefaultRoute (ecsdevops
andboxvpcPublicSubnet1DefaultRoute03AB353) andboxvpcPublicSubnet1DefaultRoute03AB353)
EcsDevopsSandboxCdkStack | 27/33 | 8:44:24 pm | CREATE_COMPLETE | AWS:EC2::RouteTableAssociation | ecs-devops-sandbox-vpc/PublicSubnet1/RouteTableAssociation (e
andboxvpcPublicSubnet1DefaultRoute03AB353) 27/33 Currently in progress: EcsDevopsSandboxCdkStack, ecsdevopsSandboxvpPublicSubnet2NATGatewayDC9E2D67, ecsdevopsSandboxserviceService73DE4E51, ecsdevopsSandboxvpcPublic
Subnet1NATGateway1261E2B8
EcsDevopsSandboxCdkStack | 28/33 | 8:45:46 pm | CREATE_COMPLETE | AWS:EC2::NatGateway | ecs-devops-sandbox-vpc/PublicSubnet2/NATGateway (ecsdevops
andboxvpcPublicSubnet1DefaultRoute03AB353) andboxvpcPublicSubnet1DefaultRoute03AB353)
EcsDevopsSandboxCdkStack | 29/33 | 8:45:48 pm | CREATE_COMPLETE | AWS:EC2::NatGateway | ecs-devops-sandbox-vpc/PublicSubnet1/NATGateway (ecsdevops
andboxvpcPublicSubnet1NATGateway1261E2B9)
EcsDevopsSandboxCdkStack | 29/33 | 8:45:48 pm | CREATE_IN_PROGRESS | AWS:EC2::Route | ecs-devops-sandbox-vpc/PrivateSubnet2/DefaultRoute (ecsdevops
andboxvpcPrivateSubnet2DefaultRoute6271FD4)
EcsDevopsSandboxCdkStack | 29/33 | 8:45:50 pm | CREATE_IN_PROGRESS | AWS:EC2::Route | ecs-devops-sandbox-vpc/PrivateSubnet2/DefaultRoute (ecsdevops
andboxvpcPrivateSubnet1DefaultRoute03AB353) andboxvpcPrivateSubnet1DefaultRoute03AB353)
EcsDevopsSandboxCdkStack | 29/33 | 8:45:50 pm | CREATE_IN_PROGRESS | AWS:EC2::Route | ecs-devops-sandbox-vpc/PrivateSubnet1/DefaultRoute (ecsdevops
sandboxvpcPrivateSubnet1DefaultRoute0393EA4F0)
EcsDevopsSandboxCdkStack | 29/33 | 8:45:51 pm | CREATE_IN_PROGRESS | AWS:EC2::Route | ecs-devops-sandbox-vpc/PrivateSubnet1/DefaultRoute (ecsdevops
sandboxvpcPrivateSubnet1DefaultRoute0393EA4F0) Resource creation Initiated
EcsDevopsSandboxCdkStack | 30/33 | 8:46:01 pm | CREATE_COMPLETE | AWS:EC2::Route | ecs-devops-sandbox-vpc/PrivateSubnet2/DefaultRoute (ecsdevops
sandboxvpcPrivateSubnet2DefaultRoute0393EA4F0) andboxvpcPrivateSubnet2DefaultRoute0393EA4F0)
EcsDevopsSandboxCdkStack | 31/33 | 8:46:01 pm | CREATE_COMPLETE | AWS:EC2::Route | ecs-devops-sandbox-vpc/PrivateSubnet1/DefaultRoute (ecsdevops
sandboxvpcPrivateSubnet1DefaultRoute0393EA4F0) andboxvpcPrivateSubnet1DefaultRoute0393EA4F0)
EcsDevopsSandboxCdkStack | 32/33 | 8:46:41 pm | CREATE_COMPLETE | AWS:ECS::Service | ecs-devops-sandbox-service/Service (ecsdevopssandboxserviceSe
rvice73DE4E51)
EcsDevopsSandboxCdkStack | 33/33 | 8:46:43 pm | CREATE_COMPLETE | AWS:CloudFormation::Stack | EcsDevopsSandboxCdkStack

EcsDevopsSandboxCdkStack
Deployment time: 231.27s
Stack ARN:
arn:aws:c:loudformation:us-east-1:689280071887:stack/EcsDevopsSandboxCdkStack/36582600-c961-11ec-8314-126c45b8aded
Total time: 242.29s
```

- ECS infrastructure built on AWS.

The screenshot shows the AWS CloudFormation console with the following details:

- Stacks**: The main navigation bar on the left.
- Stack details**: The selected category in the sidebar.
- Drifts**: A link in the sidebar.
- StackSets**: A link in the sidebar.
- Exports**: A link in the sidebar.
- Designer**: A link in the sidebar.
- Registry**: A section with links to "Public extensions", "Activated extensions", and "Publisher".
- Feedback**: A link at the bottom of the sidebar.

EcsDevOpsSandboxCdkStack (Active) - **CREATE_COMPLETE**

Overview

Stack ID	Description
arn:aws:cloudformation:us-east-1:689280071887:stack/EcsDevOpsSandboxCdkStack/36582600-c961-11ec-8314-126c45b8adef	-

Status	Status reason
CREATE_COMPLETE	-

Root stack	Parent stack
-	-

Created time	Deleted time
2022-05-01 20:43:10 UTC+0530	-

Updated time	-
2022-05-01 20:43:21 UTC+0530	-

Drift status	Last drift check time
NOT_CHECKED	-

Termination protection	IAM role
Disabled	arn:aws:iam::689280071887:role/cdk-hello-world-fifo-overwrite

Step3: Create CodeBuild Project

Screenshot of the AWS CodeBuild 'Create build project' configuration page.

Project configuration

Project name: ecs-devops-sandbox

Description - optional: (Empty text area)

Build badge - optional: Enable build badge

Enable concurrent build limit - optional: Limit the number of allowed concurrent builds for this project.
 Restrict number of concurrent builds this project can start

Additional configuration tags: (Empty text area)

Source

Source provider: GitHub

Repository: Repository in my GitHub account

GitHub repository: https://github.com/neeharika-p/ecs-devops-sandbox.git

Connection status: You are connected to GitHub using OAuth.
[Disconnect from GitHub](#)

Source version - optional info: Enter a pull request, branch, commit ID, tag, or reference and a commit ID.

Additional configuration:

- Git clone depth, Git submodules, Build status config

Git clone depth - optional: 1

Git submodules - optional: (Empty text area)

Feedback: Looking for language selection? Find it in the new Unified Settings.

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences
27°C Haze ENG IN 07:43 01-05-2022 16

The screenshot shows the 'Primary source webhook events' section of the AWS CodeBuild project configuration. It includes:

- Webhook - optional**: A checked checkbox for "Rebuild every time a code change is pushed to this repository".
- Important security considerations**: A warning icon.
- Build type**: A radio button selected for "Single build" (Triggers single build).
- Event type**: A dropdown menu with options "Start a build under these conditions" and "Don't start a build under these conditions".

The screenshot shows the "Environment" section of the AWS CodeBuild project configuration. It includes:

- Environment image**: A radio button selected for "Managed image" (Use an image managed by AWS CodeBuild).
- Operating system**: A dropdown menu set to "Ubuntu".
- Runtime(s)**: A dropdown menu set to "Standard".
- Image**: A dropdown menu set to "aws/codebuild/standard:5.0".
- Image version**: A dropdown menu set to "Always use the latest image for this runtime version".
- Environment type**: A dropdown menu set to "Linux".
- Privileged**: An unchecked checkbox for "Enable this flag if you want to build Docker images or want your builds to get elevated privileges".

A note in the operating system section states: "The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See Docker Images Provided by CodeBuild for details."

Screenshot of the AWS CodeBuild 'Create a CI/CD pipeline for your project' wizard.

Step 1: Set up build environment

Privileged: Enable this flag if you want to build Docker images or want your builds to get elevated privileges

Service role:

- New service role: Create a service role in your account
- Existing service role: Choose an existing service role from your account

Role name: codebuild-ecs-devops-sandbox-service-role

Additional configuration: Timeout, certificate, VPC, compute type, environment variables, file systems

Step 2: Buildspec

Build specifications:

- Use a buildspec file: Store build commands in a YAML-formatted buildspec file
- Insert build commands: Store build commands as build project configuration

Buildspec name - optional: By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

Step 3: Batch configuration

Project created: You have successfully created the following project: ecs-devops-sandbox

Configuration:

Source provider: GitHub	Primary repository: neeharika-p/ecs-devops-sandbox	Artifacts upload location: -	Build badge: Disabled
-------------------------	--	------------------------------	-----------------------

Build history: No results

Build history buttons: Stop build, View artifacts, View logs, Delete builds, Retry build

Build history columns: Build run, Status, Build number, Source version, Submitter, Duration, Completed

Feedback: Looking for language selection? Find it in the new Unified Settings.

System status bar: 27°C Haze, ENG IN, 07:44, 01-05-2022, 16

Screenshot of the AWS CodeBuild 'Project created' page.

Developer Tools sidebar:

- Source > CodeCommit
- Artifacts > CodeArtifact
- Build > CodeBuild
 - Getting started
 - Build projects
 - Build project**
 - Settings
 - Build history
 - Report groups
 - Report history
 - Account metrics
- Deploy > CodeDeploy
- Pipeline > CodePipeline
- Settings

Project details:

- Notify, Share, Edit, Delete build project, Start build with overrides, Start build buttons
- Configuration section: Source provider (GitHub), Primary repository (neeharika-p/ecs-devops-sandbox), Artifacts upload location (-), Build badge (Disabled)
- Build history section: Build history buttons (Stop build, View artifacts, View logs, Delete builds, Retry build), Build history table headers (Build run, Status, Build number, Source version, Submitter, Duration, Completed), Build history table body (No results)

Feedback: Looking for language selection? Find it in the new Unified Settings.

System status bar: 27°C Haze, ENG IN, 07:45, 01-05-2022, 16

Step4: Create GitHub Workflow

The screenshot shows the GitHub Actions interface. At the top, there's a search bar with the query "deploy to amazon ecs". Below it, a sidebar lists categories: Automation, Continuous integration, Deployment, and Security. The main area displays a single workflow result: "Deploy to Amazon ECS" by Amazon Web Services. It has a brief description: "Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2." Below the description are two buttons: "Configure" and "Deployment".

This screenshot shows a GitHub repository page for "neeharika-p/ecs-devops-sandbox". The user is editing the ".github/workflows/pws.yml" file. The code editor contains the following YAML configuration:

```
1 # This workflow will build and push a new container image to Amazon ECR,
2 # and then will deploy a new task definition to Amazon ECS, when there is a push to the master branch.
3 #
4 # To use this workflow, you will need to complete the following set-up steps:
5 #
6 # 1. Create an ECR repository to store your images.
7 #   For example: `aws ecr create-repository --repository-name my-ecr-repo --region us-east-2`.
8 #   Replace the value of the 'ECR_REPOSITORY' environment variable in the workflow below with your repository.
9 #   Replace the value of the 'AWS_REGION' environment variable in the workflow below with your repository's region.
10 #
11 # 2. Create an ECS task definition, an ECS cluster, and an ECS service.
12 #   For example, follow the Getting Started guide on the ECS console:
13 #   https://us-east-2.console.aws.amazon.com/ecs/home?region=us-east-2#/firstRun
14 #   Replace the value of the 'ECS_SERVICE' environment variable in the workflow below with the name you selected.
15 #   Replace the value of the 'ECS_CLUSTER' environment variable in the workflow below with the name you selected.
16 #
17 # 3. Store your ECS task definition as a JSON file in your repository.
```

To the right of the code editor, there's a Marketplace section with two actions listed: "Upload a Build Artifact" and "Setup Go environment".

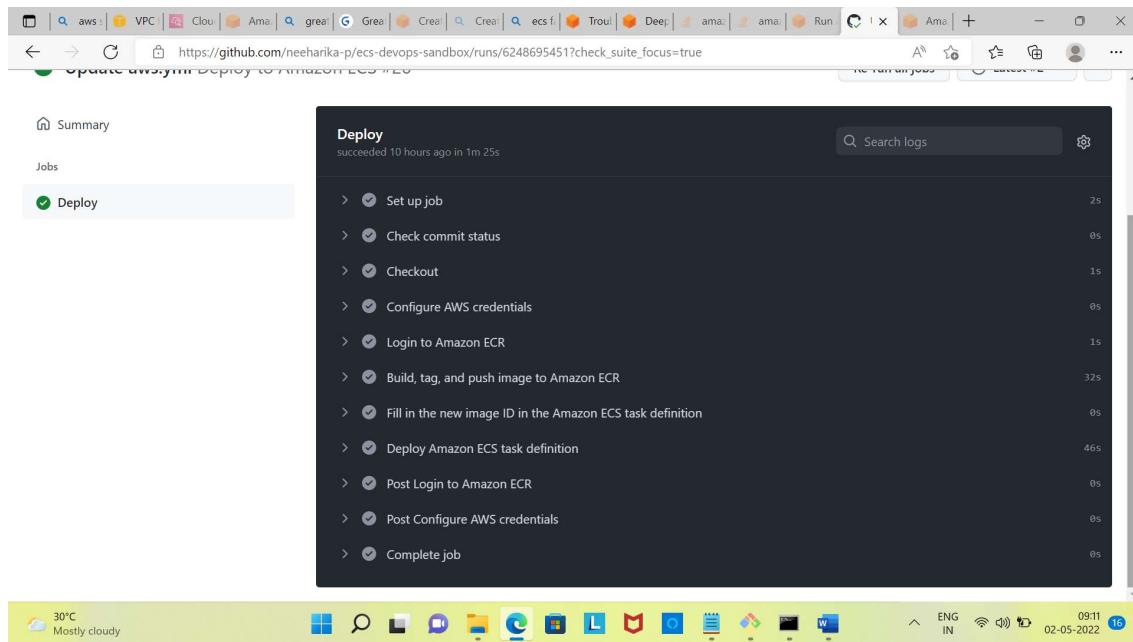
Configure AWS Credentials under Secrets

The screenshot shows the GitHub Actions secrets page for the repository 'neeharika-p/ecs-devops-sandbox'. The left sidebar is collapsed, and the main content area displays two secrets: 'AWS_ACCESS_KEY_ID' and 'AWS_SECRET_ACCESS_KEY', both updated 3 days ago. A 'New repository secret' button is visible at the top right.

Result

With the AWS CodeBuild project and GitHub workflow in place, we can automatically deploy to Amazon ECS if all tests are passed.

The screenshot shows the GitHub Actions run details for a specific job named 'Deploy' in the 'aws.yml' workflow. The job status is 'succeeded 10 hours ago in 1m 25s'. The job steps listed are: Set up job, Check commit status, Checkout, Configure AWS credentials, Login to Amazon ECR, Build, tag, and push image to Amazon ECR, Fill in the new image ID in the Amazon ECS task definition, and Deploy Amazon ECS task definition. The total execution time for these steps is 46s.



Build details from Code Build

The screenshot shows the AWS CodeBuild console. The left sidebar navigation includes "Developer Tools", "CodeBuild", "Source", "Artifacts", "Build", "Deploy", "Pipeline", and "Settings". Under "Build", "Build project" is selected. The main content area shows the "Configuration" section with "Source provider" set to GitHub and the "Primary repository" set to "neeharika-p/ecs-devops-sandbox". The "Build history" tab is active, showing a single build run:

Build run	Status	Build number	Source version	Submitter	Duration	Completed
ecs-devops-sandbox:c0b 0df49-2c7e-4895-8b60-c0db5df2649	Succeeded	1	73504a5007773de a0c233c42bbe55fd 4b01a1fca	GitHub-Hookshot/b1f 85d8	34 seconds	Just now

The browser status bar at the bottom shows the URL as https://us-east-1.console.aws.amazon.com/codesuite/codebuild/689280071887/projects/ecs-devops-sandbox/history?region=us-east-1&builds-meta=eyJmIjp7InRleHQiOiliSwicyI6e30slm4iOjlwLCpljowfQ# and the date/time as 02-05-2022 09:38.

New Container image pushed to ECR

Amazon Elastic Container Registry

Private registry

Public registry

Repositories

Summary

Images (4)

Getting started

Documentation

Public gallery

Feedback Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

35°C Mostly sunny

ENG IN 09:40 02-05-2022

Deployment event details from ECS.

Clusters

Service : ecs-devops-sandbox-service

Cluster: ecs-devops-sandbox-cluster

Status: ACTIVE

Task definition: ecs-devops-sandbox-task-definition.7

Service type: REPLICAS

Launch type: FARGATE

Service role: AWSLambdaRoleForECS

Created By: am aws iam: 689280071887:role/cdk-hnb659fds-cfn-exec-role-689280071887-us-east-1

Events

Event Id	Event Time	Message
19fcf04-4004-42d7-8705-c947c7042fa0	2022-05-02 09:40:15 +0530	service ecs-devops-sandbox-service has reached a steady state.
845995a-078-4191-9496-0937964cd0	2022-05-02 09:39:36 +0530	service ecs-devops-sandbox-service has started 1 tasks: task seeabfbaf9db3747ccbf73bc02985e25b5.
201aa867-710a-4908-8bdf-8d7f01708090	2022-05-02 09:39:03 +0530	service ecs-devops-sandbox-service has reached a steady state.
1e793a3c-ecac-4fd9-8ca1-9de8359c123	2022-05-02 09:39:03 +0530	service ecs-devops-sandbox-service (deployment ecs-svc/226492313931548800) deployment completed.
da204ba-400f-4371-8749-979cfed947	2022-05-02 09:38:24 +0530	service ecs-devops-sandbox-service has started 1 tasks: task 9bdcff7b7f3743bbaefdb8370d405344.

Feedback: Looking for language selection? Find it in the new [Unified Settings](#)

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

35°C Mostly sunny

ENG IN 09:41 02-05-2022

Cost Analysis

Code Build	100 builds per month x 1 minutes = 100.00 billed minutes (monthly) 100.00 minutes x 0.005 USD = 0.50 USD AWS CodeBuild cost (monthly): 0.50 USD
Cloud watch logs	Data Ingested to cloud watch logs not more than 1GB Cloud watch logs cost (monthly): 1USD
AWS Fargate ECS Pricing	Number of tasks or pods: 1 per minute * (60 minutes in an hour x 730 hours in a month) = 43800 per month

Average duration: 1 minutes = 0.02 hours
Pricing calculations
43,800 tasks x 1 vCPU x 0.02 hours x 0.04048 USD per hour = 35.46 USD for vCPU hours
43,800 tasks x 2.00 GB x 0.02 hours x 0.004445 USD per GB per hour = 7.79 USD for GB hours
20 GB - 20 GB (no additional charge) = 0.00 GB billable ephemeral storage per task
35.46 USD for vCPU hours + 7.79 USD for GB hours = 43.25 USD total
Fargate cost (monthly): 43.25 USD

Total Solution cost = 45 USD