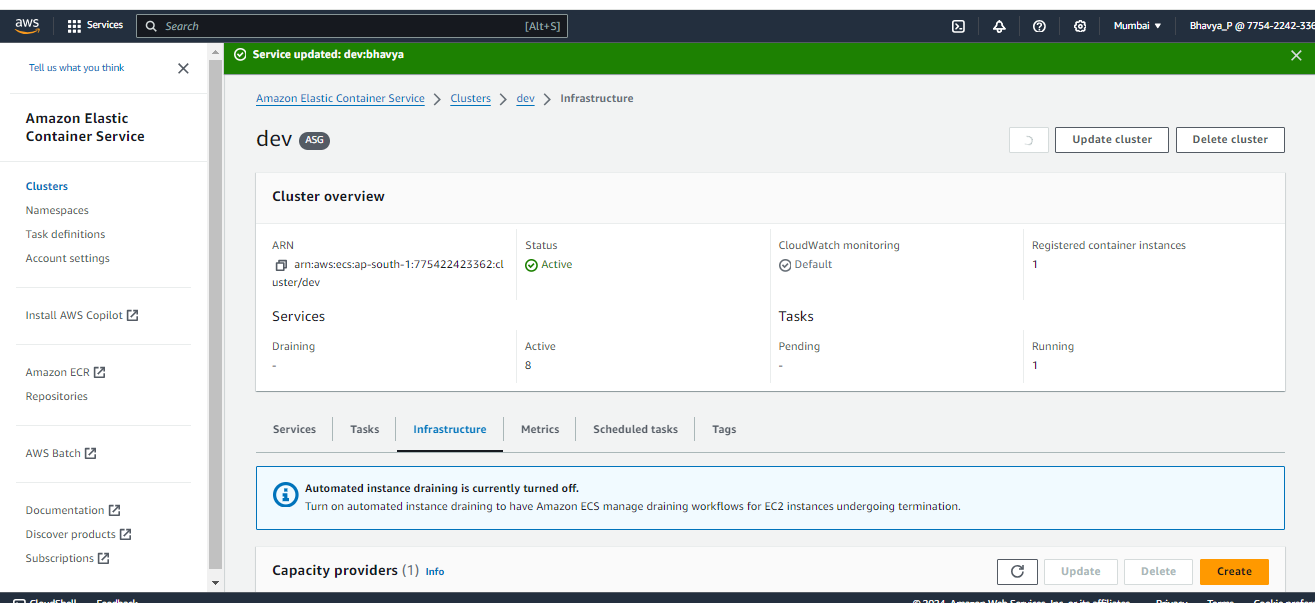
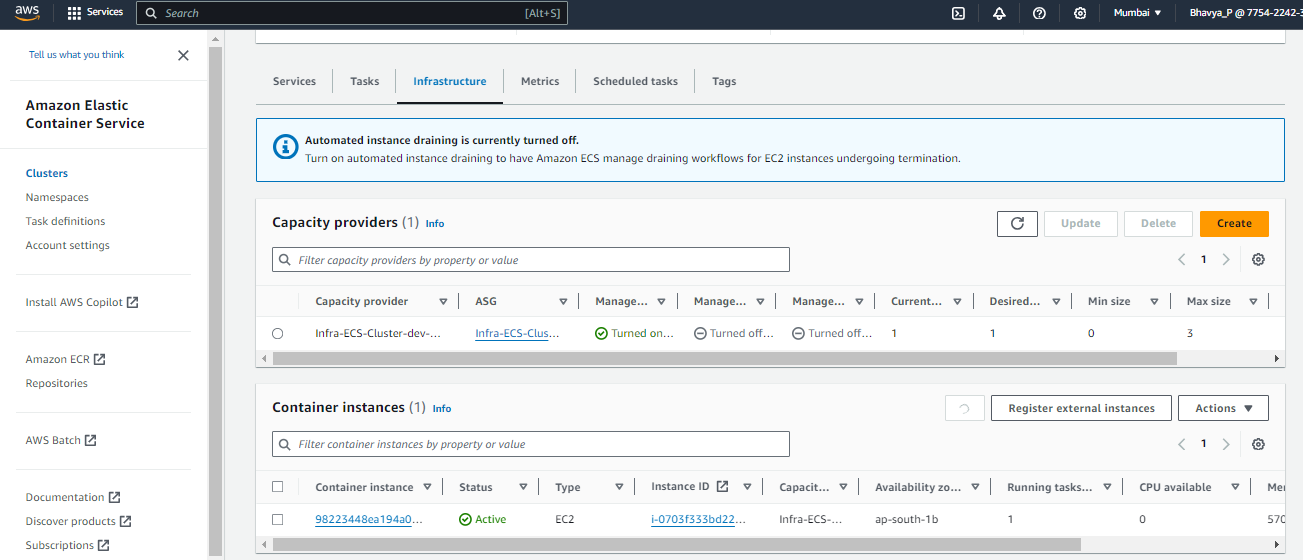
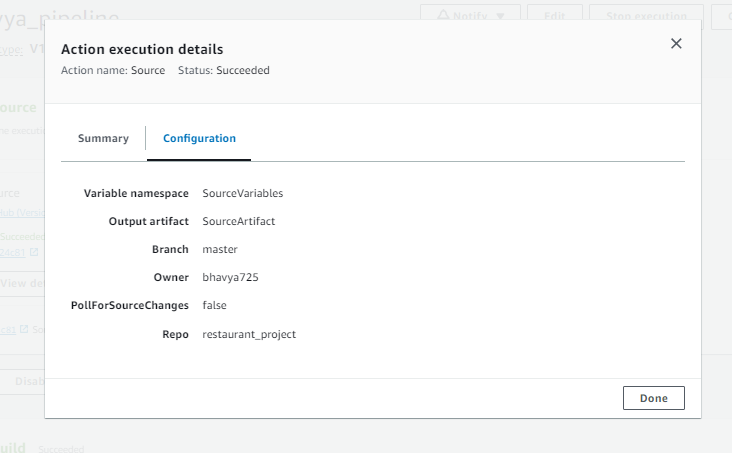
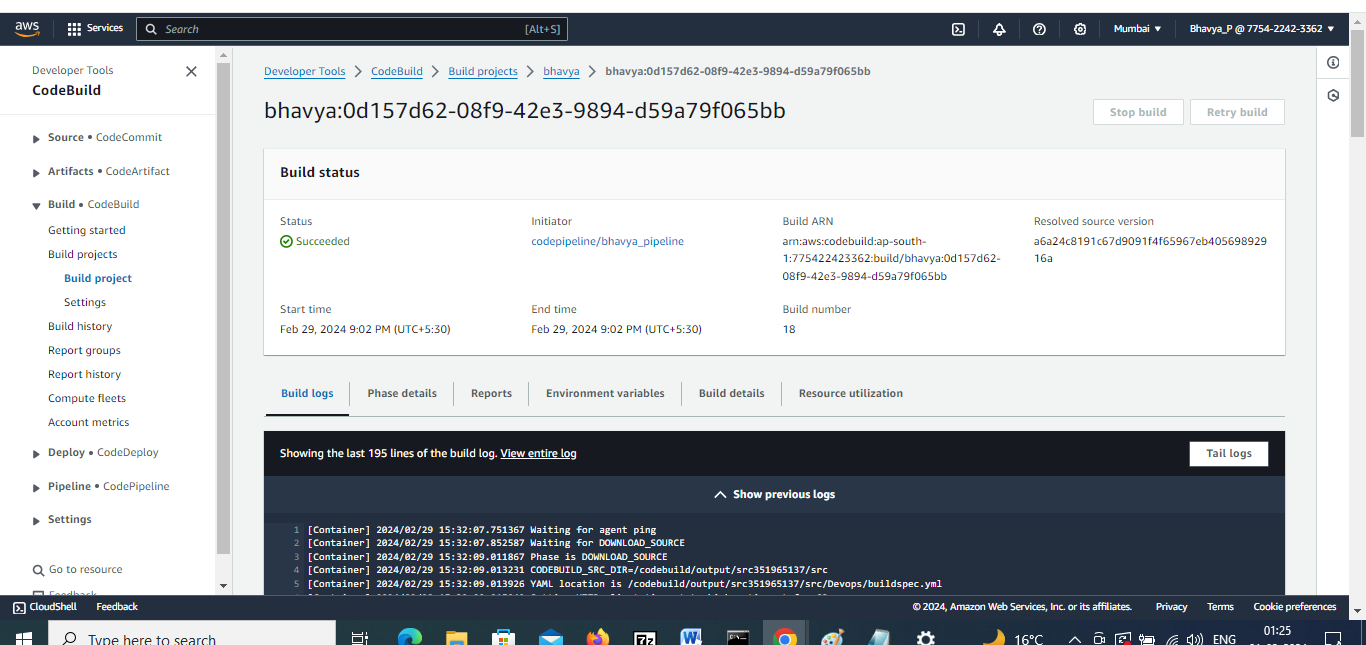
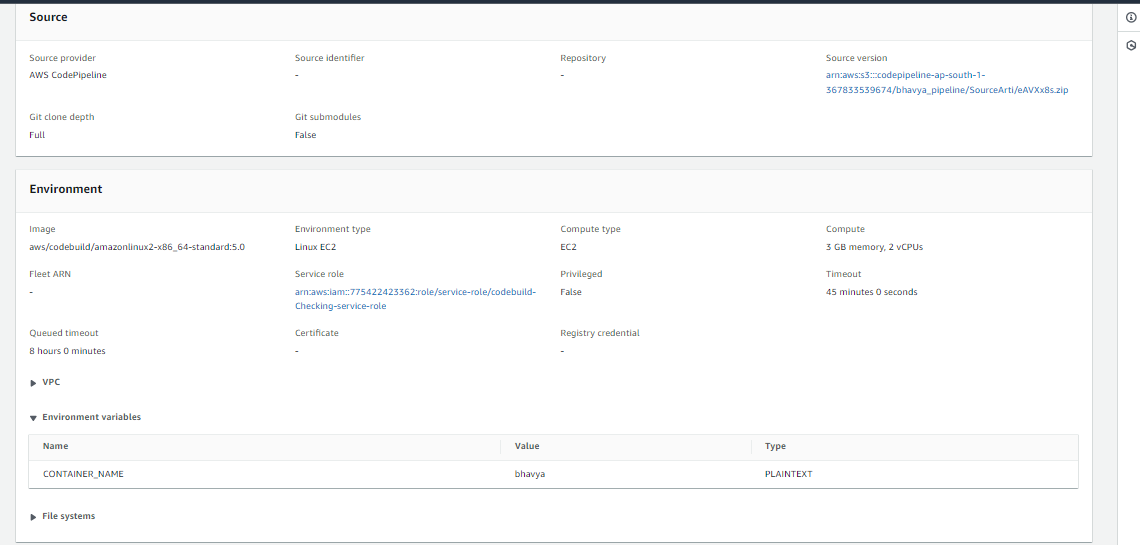
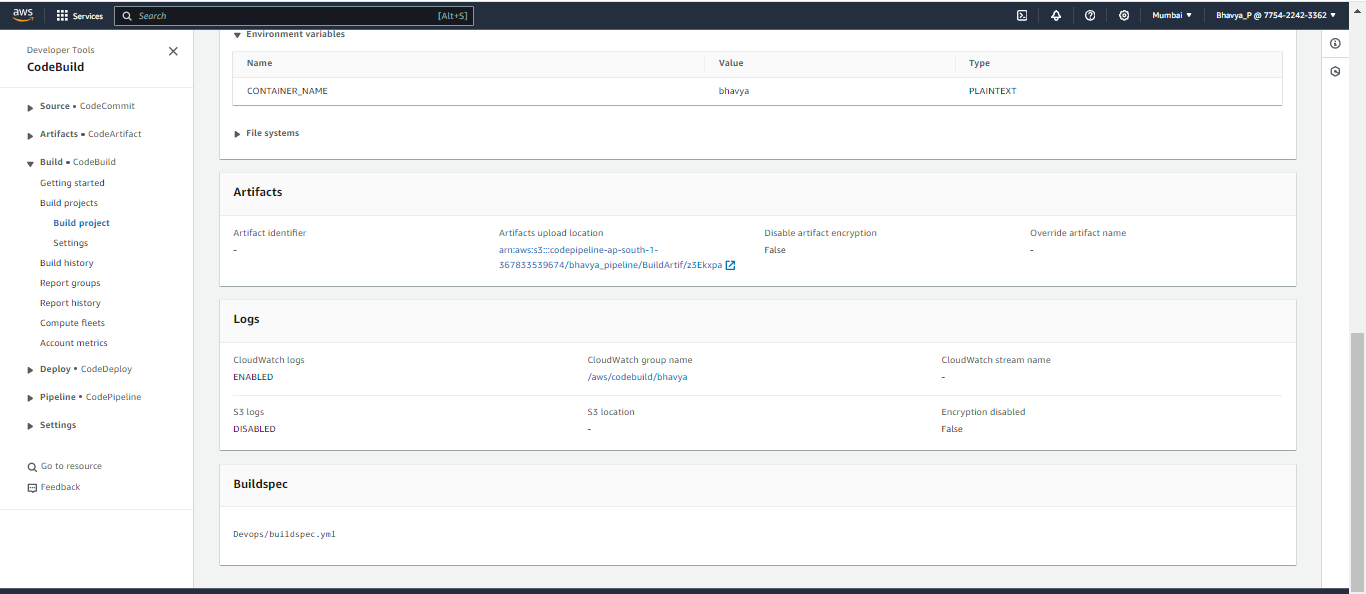
**Creating a Code pipeline to deploy your application through ECS as and when there is any change pushed from your github repo**

Create an ECS cluster fro deployment



Attach that cluster to you ASG  
  
and make the desired capacity to 1, when you want to run your service in a task.  
  
Create task definition, with ec2 as launch type , add the container name with image URI maped to a container port with a protocol attached to it  
Create a service with the task definition created above and run the task out of it,  
  
Now, check the ip address of the container with the port number mapped on container.  
  
code pipeline:  
  
  
create a pipeline, from code commit taking github v1 as source, authenticate your github username and password, then choose your project repo in github  
  
Execution mode: suspended and pipeline type: v1  
  
here in this path you’ll have your Dockerfile and buildspec.yml files so that code build would build your image, and execute build commands before deploying your application.  
  
  
  
  
  
  
  
while creating pipeline, an s3 bucket would be created, which would store the artifacts  
  
need to specify a buildspec.yml file with the location of file in github.

--------------------------------------------------------------------------------------------------------------------------  
  
version: 0.2

phases:

pre\_build:

commands:

- echo Logging in to Amazon ECR...

- docker --version

- aws --version

- AWS\_DEFAULT\_REGION=ap-south-1

- AWS\_ACCOUNT\_ID=775422423362

- IMAGE\_REPO\_NAME=bhavya

- docker login -u AWS -p $(aws ecr get-login-password --region ap-south-1) 775422423362.dkr.ecr.ap-south-1.amazonaws.com

# - aws ecr get-login-password --region $AWS\_DEFAULT\_REGION | docker login --username AWS --password-stdin $AWS\_ACCOUNT\_ID.dkr.ecr.$AWS\_DEFAULT\_REGION.amazonaws.com

- COMMIT\_HASH=$(echo $CODEBUILD\_RESOLVED\_SOURCE\_VERSION | cut -c 1-7)

# - IMAGE\_TAG=build=$(echo $CODEBUILD\_BUILD\_ID | awk -F":" '{print $2}')

- IMAGE\_TAG=$(echo $CODEBUILD\_BUILD\_ID | awk -F":" '{print $2}')

# - IMAGE\_TAG=latest

- REPOSITORY\_URI=$AWS\_ACCOUNT\_ID.dkr.ecr.$AWS\_DEFAULT\_REGION.amazonaws.com/$IMAGE\_REPO\_NAME

build:

commands:

- echo Build started on `date`

- echo Building the Docker image...

- docker build -t $REPOSITORY\_URI:$IMAGE\_TAG .

- docker tag $REPOSITORY\_URI:$IMAGE\_TAG $REPOSITORY\_URI:$IMAGE\_TAG

post\_build:

commands:

- echo Build completed on `date`

- echo Pushing the Docker images...

- docker push $REPOSITORY\_URI:$IMAGE\_TAG

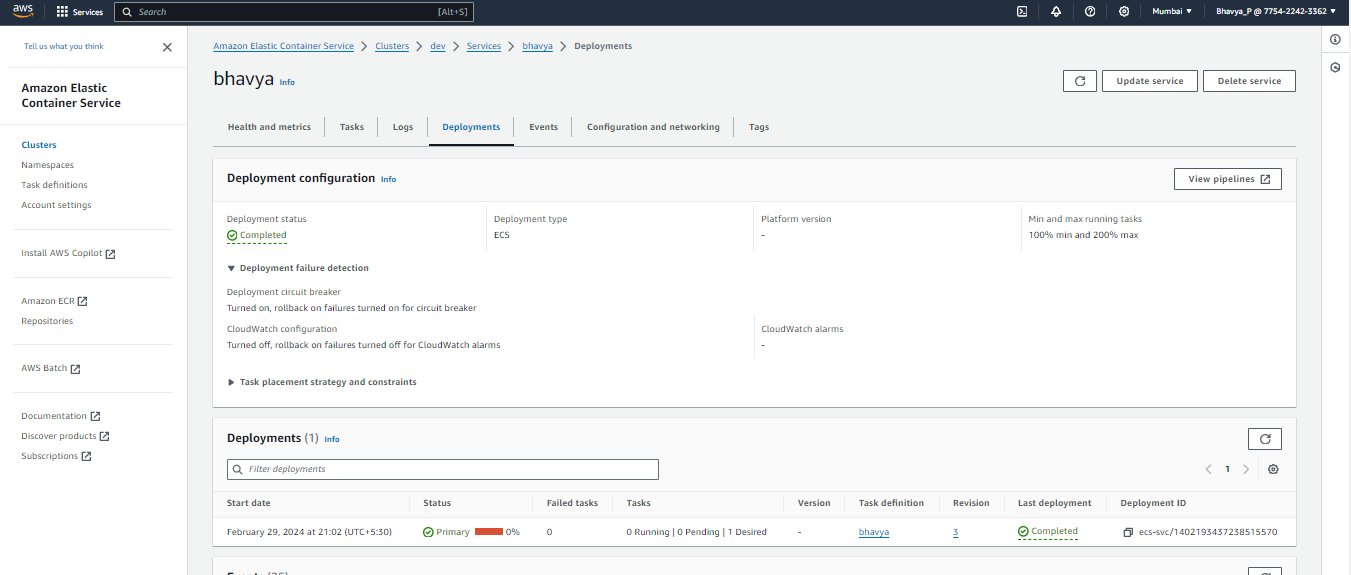
- echo Writing image definitions file...

- printf '[{"name":"%s","imageUri":"%s"}]' $CONTAINER\_NAME $REPOSITORY\_URI:$IMAGE\_TAG > imagedefinitions.json

artifacts:

files:

- imagedefinitions.json

------------------------------------------------------------------------------------------------------------------------------------  
  
Now , choose your deployments from ECS as we have deployed our app there  
  
  
  
This would inturn run your pipeline, as and when there would be any push done in your github